

## Documento de pruebas realizadas a los endpoints y retrospectiva del Sprint

Equipo:	2 (Jhon Lerma, Carlos Lozano, Andres Gonzalez, David Quinayas, Paula Rosero)
Sprint No.:	1
Github link	<a href="https://github.com/jhonlerma/RegistrApp_results_backend">https://github.com/jhonlerma/RegistrApp_results_backend</a>

1. A continuación se presentan las capturas de pantalla de las pruebas realizadas a todos los endpoints.

#### **a. Administración de Candidatos**

## Creación de Candidato

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main workspace is titled "My Workspace" and contains a collection named "RegistraAppResultBackend". This collection includes several endpoints: "candidate/list" (GET), "candidate/(id)" (GET), "candidate/update" (PUT), "candidate/create" (POST), and "candidate/delete" (DELETE). The "candidate/create" endpoint is currently selected. The "Body" tab is active, showing a JSON payload:

```
1 "document": "1234",
2 "last_name": "carlos",
3 "name": "lozano",
4 "resolution": "123456"
```

Below the body, the "Test Results" section shows a successful response with status 201 CREATED, time 124 ms, and size 241 B. The response body is also displayed:

```
1 "id": "630fe5ad8fd0e21716786fac"
```

## Listado de Candidatos

The screenshot shows the Postman interface with the 'Listado de Candidatos' (Candidate List) API endpoint selected. The request URL is `http://localhost:3000/candidate/list`. The response body is a JSON array containing two candidate objects:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
{
  "_id": "638fabd906850bc9f22669f5",
  "document": "123",
  "last_name": "leirma",
  "name": "jhon",
  "resolution": "1345"
},
{
  "_id": "638fe5ad8fd0e21716786fac",
  "document": "1234",
  "last_name": "carlos",
  "name": "lozano",
  "resolution": "123456"
}
```

## Buscar Candidato por ID

The screenshot shows the Postman interface with the 'Buscar Candidato por ID' (Search Candidate by ID) API endpoint selected. The request URL is `http://localhost:3000/candidate/630fabd906850bc9f22669f5`. The response body is a single candidate object:

```
1
2
3
4
5
6
7
{
  "_id": "630fabd906850bc9f22669f5",
  "document": "123",
  "last_name": "leirma",
  "name": "jhon",
  "resolution": "1345"
}
```

## Actualización de Candidato

The screenshot shows the Postman interface with the following details:

- Collection:** ResgistraAppResultBackend
- Request:** PUT /candidate/{id}
- Method:** PUT
- URL:** http://localhost:3000/candidate/630fabd906850bc9f22669f5
- Body:** JSON (Pre-request Script: `document = "12345";`)
- Status:** 204 NO CONTENT
- Time:** 126 ms
- Size:** 185 B

The screenshot shows the Postman interface with the following details:

- Collection:** ResgistraAppResultBackend
- Request:** GET /candidate/{id}
- Method:** GET
- URL:** http://localhost:3000/candidate/630fabd906850bc9f22669f5
- Headers:** (6)
- Query Params:** (1)

KEY	VALUE	DESCRIPTION
Key	Value	Description
- Status:** 200 OK
- Time:** 118 ms
- Size:** 326 B

## Eliminar Candidato

The screenshot shows the Postman interface with the following details:

- Collection:** ResgistraAppResultBackend
- Request Type:** DELETE
- URL:** http://localhost:3000/candidate/630fe5ad8fd0e21716786fac
- Headers:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings
- Body:** Query Params
- Response Status:** 204 NO CONTENT
- Time:** 139 ms
- Size:** 185 B

The screenshot shows the Postman interface with the following details:

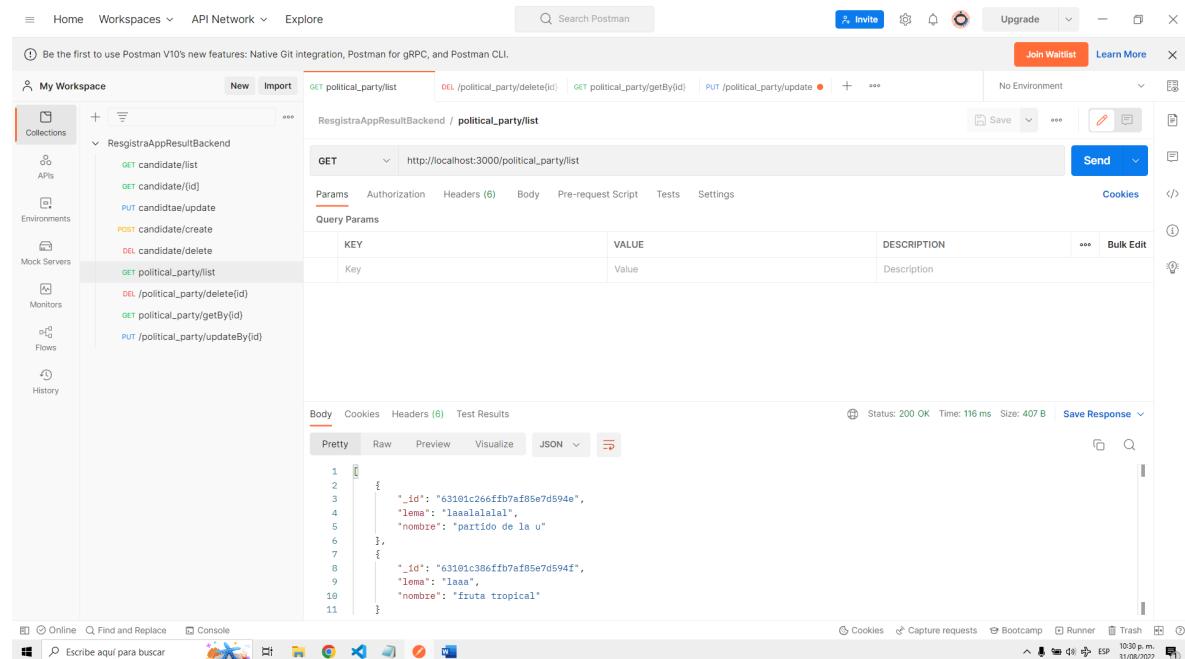
- Collection:** ResgistraAppResultBackend
- Request Type:** GET
- URL:** http://localhost:3000/candidate/list
- Headers:** Authorization, Headers (6), Body, Pre-request Script, Tests, Settings
- Body:** Query Params
- Response Status:** 200 OK
- Time:** 115 ms
- Size:** 344 B

The response body is a JSON array:

```
[{"_id": "630fabd9868560bc9f22669f5", "document": "1235", "last_name": "leme", "name": "jhon", "resolution": "1345"}]
```

## b. Administración de Partidos

### Listado de Partidos

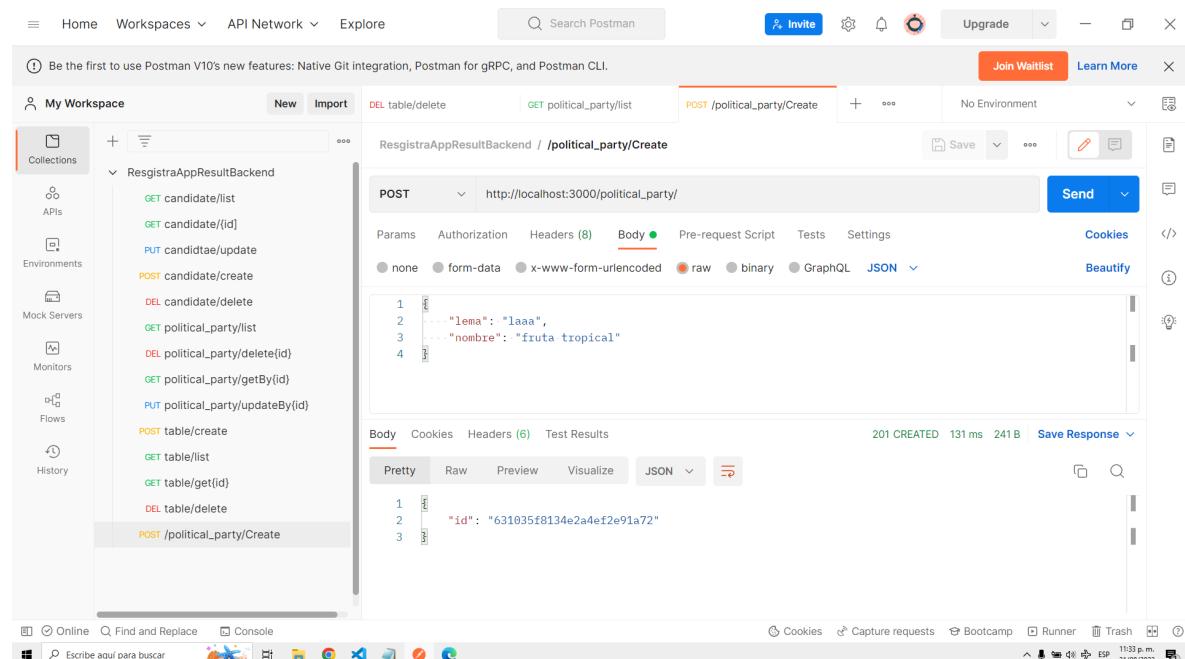


The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Upgrade, Join Waitlist, Learn More.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request List:** RegistrAppResultBackend / political\_party/list (highlighted).
  - Method: GET
  - URL: http://localhost:3000/political\_party/list
  - Params tab (selected)
  - Headers tab
  - Body tab
  - Pre-request Script
  - Tests
  - Settings
- Response Preview:** Status: 200 OK, Time: 116 ms, Size: 407 B, Save Response.
  - Body tab (selected): Pretty, Raw, Preview, Visualize, JSON.
  - Content:

```
1 {
2   "_id": "63101c26effb7af85e7d594e",
3   "lema": "laalalalal",
4   "nombre": "partido de la u"
5 },
6 {
7   "_id": "63101c386ff7af85e7d594f",
8   "lema": "laaa",
9   "nombre": "fruta tropical"
10 }
```
- Bottom Bar:** Online, Find and Replace, Console, Escribe aquí para buscar, Windows taskbar.

### Creación de Partido



The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Upgrade, Join Waitlist, Learn More.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request List:** RegistrAppResultBackend / political\_party/Create (highlighted).
  - Method: POST
  - URL: http://localhost:3000/political\_party/
  - Params tab
  - Authorization tab
  - Headers tab (selected)
  - Body tab (selected)
    - none
    - form-data
    - x-www-form-urlencoded
  - Pre-request Script
  - Tests
  - Settings
- Response Preview:** Status: 201 CREATED, Time: 131 ms, Size: 241 B, Save Response.
  - Body tab (selected): Pretty, Raw, Preview, Visualize, JSON.
  - Content:

```
1 {
2   "_id": "631035f8134e2a4ef2e91a72"
3 }
```
- Bottom Bar:** Online, Find and Replace, Console, Escribe aquí para buscar, Windows taskbar.

## Buscar Partido por ID

The screenshot shows the Postman interface with the following details:

- URL:** http://localhost:3000/political\_party/63101c266ffb7af85e7d594e
- Method:** GET
- Headers:** (6)
- Body:** (empty)
- Response:**

```
1  {
2     "_id": "63101c266ffb7af85e7d594e",
3     "lema": "laalalalalal",
4     "nombre": "partido de la u"
```

## Actualización de Partido por ID

The screenshot shows the Postman interface with the following details:

- URL:** http://localhost:3000/political\_party/63101c266ffb7af85e7d594e
- Method:** PUT
- Headers:** (8)
- Body:** (raw)

```
1 {
2     "lema": "Para vivir sabroso"
3 }
```
- Response:**

```
1
2 Status: 204 NO CONTENT
```

Home Workspaces API Network Explore

Search Postman

Join Waitlist Learn More

My Workspace

New Import GET political\_party/list DEL /political\_party/delete{id} GET political\_party/getBy{id} PUT /political\_party/update

ResgistraAppResultBackend / political\_party/getBy{id}

GET http://localhost:3000/political\_party/63101c266fffb7af85e7d594e

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Bulk Edit

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1
2     "id": "63101c266fffb7af85e7d594e",
3     "lema": "Para vivir sabroso",
4     "nombre": "partido de la u"
```

Status: 200 OK Time: 150 ms Size: 301 B Save Response

Collections APIs Environments Mock Servers Monitors Flows History

Online Find and Replace Console Escribe aqui para buscar

Cookies Capture requests Bootcamp Runner Trash 1031 p.m. 31/08/2022

## Eliminar Partido por ID

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Upgrade, Join Waitlist, Learn More.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request Details:** ResgistraAppResultBackend / political\_party/delete{id}
- Method:** DELETE
- URL:** http://localhost:3000/political\_party/63101c386ffb7af85e7d594f
- Params:** (empty)
- Headers:** (6)
- Body:** (empty)
- Test Results:** 204 NO CONTENT, 139 ms, 185 B
- Buttons:** Send, Save Response.

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Upgrade, Join Waitlist, Learn More.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request Details:** ResgistraAppResultBackend / political\_party/list
- Method:** GET
- URL:** http://localhost:3000/political\_party/list
- Params:** (empty)
- Headers:** (6)
- Body:** (empty)
- Test Results:** 200 OK, 131 ms, 300 B
- Buttons:** Send, Save Response.

### c. Administración de mesas

## Listado de Mesas

The screenshot shows the Postman application interface. On the left, the sidebar includes sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows, and History. The main workspace displays a collection named "ResgistraAppResultBackend" which contains several endpoints: GET candidate/ist, GET candidate/{id}, PUT candidatæ/update, POST candidate/create, DELETE candidate/delete, GET political\_party/ist, DELETE political\_party/delete{id}, GET political\_party/getBy{id}, PUT political\_party/updateBy{id}, POST political\_party/Create, POST table/create, GET table/list, GET table/get{id}, and DELETE table/delete. The "table/list" endpoint is currently selected. The request details panel shows a GET request to "http://localhost:3000/table/list". The "Body" tab is active, displaying a JSON response with two objects:

```
1 {  
2   "_id": "631030b6624dae6a0d260a77",  
3   "cantidad_inscritos": 30,  
4   "numero": 12  
5 },  
6 {  
7   "_id": "631032b9624dae6a0d260a78",  
8   "cantidad_inscritos": 40,  
9   "numero": 17  
10 },  
11 }
```

The "Tests" tab indicates 200 OK, 107 ms, and 486 B. The bottom navigation bar includes links for Online, Find and Replace, Console, Cookies, Capture requests, Bootcamp, Runner, Trash, and Help.

## Creación de Mesa

## Buscar Mesa por ID

The screenshot shows the Postman application interface. On the left, there's a sidebar titled "My Workspace" with various collections like "ResgistraAppResultBackend". The main area displays a "RegistraAppResultBackend / table/get{id}" collection. A specific GET request is selected, which has the URL `http://localhost:3000/table/631030b6624dae6a0d260a77`. The "Body" tab is active, showing a JSON response with fields: `_id": "631030b6624dae6a0d260a77", "cantidad_inscritos": 30, "numero": 12`. The status bar at the bottom indicates a 200 OK response with 126 ms and 281 B.

## Actualización de Mesa por ID

The screenshot shows the Postman application interface again. The "My Workspace" sidebar is visible. A "PUT table/update" request is selected, with the URL `http://localhost:3000/table/631032b8624dae6a0d260a78`. The "Body" tab is active, showing a JSON payload with the field `"cantidad_inscritos": 480`. The status bar at the bottom indicates a 204 NO CONTENT response with 151 ms and 185 B.

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Notifications, Upgrade.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request Details:** ResgistraAppResultBackend / table/get{id}, GET method, http://localhost:3000/table/631032b8624dae6a0d260a78.
- Query Params:** Key, Value, Description.
- Body:** JSON response:

```
1  {
2      "_id": "631032b8624dae6a0d260a78",
3      "cantidad_inscriptos": 488,
4      "numero": 17
5 }
```
- Status:** 200 OK, 131 ms, 282 B.
- Bottom Bar:** Cookies, Capture requests, Bootcamp, Runner, Trash, Date: 11:41 p.m. 31/08/2022.

## Eliminar Mesa por ID

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Notifications, Upgrade.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request Details:** ResgistraAppResultBackend / table/delete, DELETE method, http://localhost:3000/table/631030b6624dae6a0d260a77.
- Query Params:** Key, Value, Description.
- Body:** JSON response:

```
1
```
- Status:** 204 NO CONTENT, 123 ms, 185 B.
- Bottom Bar:** Cookies, Capture requests, Bootcamp, Runner, Trash, Date: 11:41 p.m. 31/08/2022.

The screenshot shows the Postman interface with the following details:

- Request:** GET http://localhost:3000/table/list
- Body:** JSON
- Response:**

```

1 [
2   {
3     "_id": "631032b8624dae6a0d260a78",
4     "cantidad_inscriptos": 40,
5     "numero": 17
6   },
7   {
8     "_id": "63103655134e2a4ef2e91a73",
9     "cantidad_inscriptos": 40,
10    "numero": 18
11 }

```

## d. Administración de resultados

### Listado de Resultados

The screenshot shows the Postman interface with the following details:

- Request:** GET http://localhost:3000/result/list
- Body:** JSON
- Response:**

```

1 [
2   {
3     "_id": "63103a87da8eb254e808de46",
4     "id_partido": "52",
5     "numero_mesa": "20"
6   },
7   {
8     "_id": "63103b25da8eb254e808de47",
9     "id_partido": "25",
10    "numero_mesa": "20"
11 }

```

## Creación de Resultado

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing various API endpoints like 'PUT candidatae/update', 'POST candidate/create', etc. The main area displays a POST request to 'http://localhost:3000/result'. The 'Body' tab is selected, showing a JSON payload:

```
1 ... "numero_mesa": "10",
2 ...
3 ...
4 ...
5 ...
6 ...
7 ...
8 ...
9 ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
```

The response status is 201 CREATED with a response time of 126 ms and a size of 241 B. The response body is also shown in JSON format:

```
1 ...
2 ...
3 ...
4 ...
5 ...
6 ...
7 ...
8 ...
9 ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
```

This screenshot shows another view of the Postman interface. It features a similar sidebar with various API endpoints. The main area displays a GET request to 'http://localhost:3000/result/list'. The 'Headers' tab is selected, showing 'Content-Type: application/json'. The 'Body' tab is selected, showing a table for 'Query Params' with one row: 'Key' and 'Value'. The response status is 200 OK with a response time of 109 ms and a size of 489 B. The response body is shown in JSON format:

```
1 ...
2 ...
3 ...
4 ...
5 ...
6 ...
7 ...
8 ...
9 ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
```

## Buscar Resultado por ID

The screenshot shows the Postman application interface. On the left, there's a sidebar with various sections like Home, Workspaces, API Network, and Explore. The main area is titled "My Workspace" and contains a collection of API endpoints. One endpoint is selected: "GET result/getBy{id}" from the "RegistraAppResultBackend" collection. The request details show a GET method with the URL `http://localhost:3000/result/63103ba3da8eb254e808de48`. The "Body" tab is selected, displaying a JSON response:

```
1  {
2   "id": "63103ba3da8eb254e808de48",
3   "id_partido": "15",
4   "numero_mesa": "10"
5 }
```

The status bar at the bottom indicates a 200 OK response with 109 ms latency and 282 B size.

## Actualización de Resultado por ID

The screenshot shows the Postman interface with the following details:

- Request Type:** PUT
- URL:** `http://localhost:3000/result/63103ba3da8eb254e808de48`
- Body:** `"id_partido": "102"`
- Response Status:** 204 NO CONTENT
- Time:** 120 ms
- Size:** 185 B

The screenshot shows the Postman interface with the following details:

- Request Type:** GET
- URL:** `http://localhost:3000/result/63103ba3da8eb254e808de48`
- Headers:** (6)
- Body:** `{"_id": "63103ba3da8eb254e808de48", "id_partido": "102", "numero_mesa": "10"}`
- Response Status:** 200 OK
- Time:** 109 ms
- Size:** 283 B

## Eliminar Resultado por ID

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Settings, Upgrade, Join Waitlist, Learn More.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request Details:** ResgistraAppResultBackend / result/delete, Method: DELETE, URL: http://localhost:3000/result/63103ba8eb254e808de48. Params tab selected.
- Body:** Body tab selected, showing a single key "Key" with value "Value".
- Response:** Status: 204 NO CONTENT, Time: 105 ms, Size: 185 B.
- Bottom:** Cookies, Capture requests, Bootcamp, Runner, Trash, Date: 11:59 p.m. 31/08/2022.

The screenshot shows the Postman interface with the following details:

- Header Bar:** Home, Workspaces, API Network, Explore, Search Postman, Invite, Settings, Upgrade, Join Waitlist, Learn More.
- Left Sidebar:** My Workspace, Collections, APIs, Environments, Mock Servers, Monitors, Flows, History.
- Request Details:** ResgistraAppResultBackend / result/list, Method: GET, URL: http://localhost:3000/result/list. Params tab selected.
- Body:** Body tab selected, showing a JSON response with two objects:

```
1  [
2    {
3      "_id": "63103a87da8eb254e808de46",
4      "id_partido": "52",
5      "numero_mesa": "20"
6    },
7    {
8      "_id": "63103b25da8eb254e808de47",
9      "id_partido": "25",
10     "numero_mesa": "20"
11   }
12 ]
```

- Response:** Status: 200 OK, Time: 110 ms, Size: 393 B.
- Bottom:** Cookies, Capture requests, Bootcamp, Runner, Trash, Date: 12:00 a.m. 1/09/2022.

**2. Captura de pantalla del tablero de JIRA donde se evidencie las tareas realizadas por cada miembro del grupo.**

The screenshot shows a JIRA board titled "Tablero Sprint 1". The board has three columns: "POR HACER", "EN CURSO", and "LISTO". The "LISTO" column is expanded, showing a list of tasks with their status and assignees:

- Cambio de Roles (DE)
- Implementar Endpoint "Administración de Resultados" (DE)
- Implementar Endpoint "Administración de Partidos" (PP)
- Implementar Endpoint "Administración de Candidatos" (JB)
- Implementar Endpoint "Administración de Mesas" (AP)
- Realizar pruebas de cada uno los Endpoints (CT)
- Subir plantilla repositorio en GitHub (CT)
- Crear Documento de Balance Final del Sprint (AP)

Each task in the "LISTO" column has a checkmark next to it, indicating they have been completed.

**3. Retrospectiva del Sprint**

a. Lo que se logró:

- Fue posible el desarrollo y prueba de todos los “endpoints”.
- Se realizó una implementación adecuada de las carpetas en el repositorio.
- Se logró una buena integración entre las diferentes plataformas y los miembros del equipo de trabajo.
- Con base en las pruebas realizadas se obtuvieron resultados satisfactorios en cada una de ellas.

b. A mejorar:

- Tener claridad con las fechas para realizar una mejor gestión de tiempo para llevar a cabo satisfactoriamente las tareas de cada Sprint y distribuir uniformemente el trabajo por unidad de tiempo.

