ESPECIFICACIONES TÉCNICAS

Backend de clientes (CRUD Clientes)

Sprint - Deadline	**2 - 3/11/23**
Jira asociado	**SCRUM-25**
Responsable	**Javier Huebra**
Equipo	• Miguel Ángel Maisares • Lucas Sarfati • Javier Huebra • Matías Stewart Usher • Facundo Oliva • Koba Chajchir • Yamil Leotta
Versión	1.0

Contexto

Se realiza el CRUD de Clientes respetando los patrones iniciales para este tipo de arquitectura.

Objetivo

Crear el backend completo que relaciona a **Personas** y **Empresas** bajo la entidad **Clientes** y que permita Crear, Modificar, Eliminar y Actualizar los valores.

Alcance

- Crear un nuevo Cliente utilizando un único endpoint de creación.
- Modificar un Cliente, se puede cambiar tanto su tipo como la Persona y/o Empresa que lo conforman..
- Alternar estado de cliente (Eliminación lógica).
- Listar un cliente por id.
- Listar todos los Clientes registrados.

Entregable

Endpoint de creación: POST /api/clientes

- **Tipo Empresa:**
  - Ejemplo de body request correcto:

```
{
    "persona_id": 1,
    "empresa_id": 1
}
```

  - Ejemplo de body response correcto **(status 201)** :

```
{
    "id": 1,
    "persona": {
        "id": 1,
        "nombre": "John",
        "apellido": "Doe",
        "dni": "40358987",
        "eliminado": false
    },
    "empresa": {
        "id": 1,
        "nombre": "Empresa uno",
        "cuit": "1231231",
        "start_date": "2024-10-27T15:30:00"
    },
    "eliminado": false
}
```

- **Tipo Persona:**
  - Ejemplo de body request correcto:

```
{
    "persona_id": 1,
    "empresa_id": null
}
```

  - Ejemplo de body response correcto **(201)**:

```
{
    "id": 2,
    "persona": {
        "id": 1,
        "nombre": "John",
        "apellido": "Doe",
        "dni": "40358987",
        "eliminado": false
    },
    "empresa": null,
    "eliminado": false
}
```

Si se coloca un valor no nulo en ambos ID se está creando un cliente de tipo **EMPRESA.** Por otro lado, si el id de empresa es `null` se interpretará como la creacion de un cliente tipo **PERSONA.**

Validaciones de la creación de un cliente

Si el json no respeta el formato (se envía de forma errónea) → 400 Bad Request

Si ya existe el cliente de tipo persona que se intenta crear → 400 Bad Request

Si ya existe el cliente de tipo empresa que se intenta crear → 400 Bad Request

Si el json posee el id de la persona en null → 400 Bad Request

Si no encuentra a la persona o la empresa por el id que se manda en el json → 404 Not Found

Endpoint para listar un cliente: GET /api/clientes/{id}

- No se requiere body request.
- Ejemplo de body response OK **(status 200)**:

```
{
    "id": 1,
    "persona": {
        "id": 1,
        "nombre": "John",
        "apellido": "Doe",
        "dni": "40358987",
        "eliminado": false
    },
    "empresa": {
        "id": 1,
        "nombre": "Empresa uno",
        "cuit": "1231231",
        "start_date": "2024-10-27T15:30:00"
    },
    "eliminado": false
}
```

Validaciones de la obtención de un cliente

Si el id enviado por el parametro url no respeta el formato numérico → 400 Bad Request

Si no se encuentra un cliente bajo ese numero de identificación → 404 Not Found

Endpoint para listar todos los clientes: GET /api/clientes

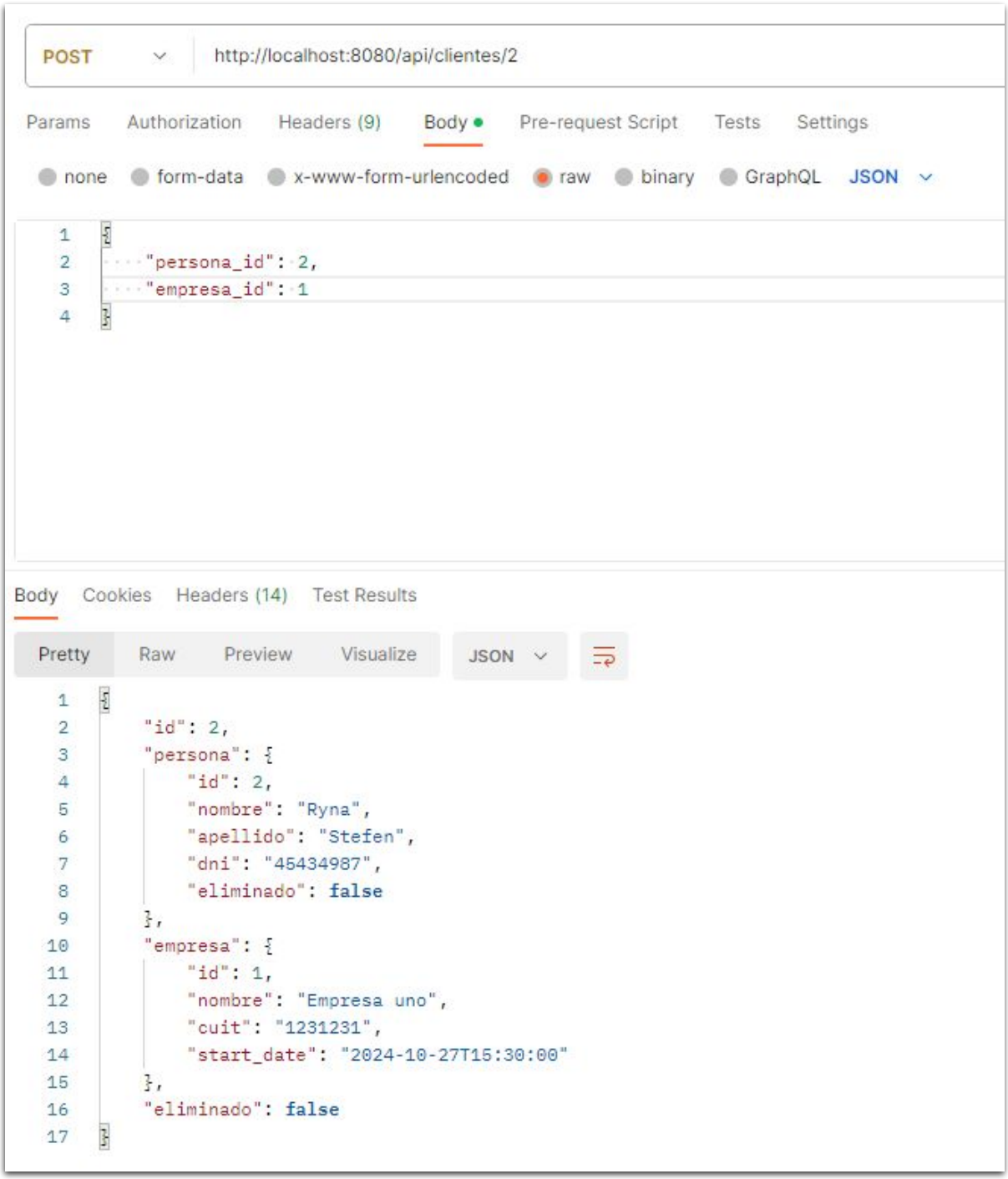
- Ejemplo de body response OK **(status 200)**:

```
[
    {
        "id": 1,
        "persona": {
            "id": 1,
            "nombre": "John",
            "apellido": "Doe",
            "dni": "40358987",
            "eliminado": false
        },
        "empresa": {
            "id": 1,
            "nombre": "Empresa uno",
            "cuit": "1231231",
            "start_date": "2024-10-27T15:30:00"
        },
        "eliminado": false
    },
    {
        "id": 2,
        "persona": {
            "id": 1,
            "nombre": "John",
            "apellido": "Doe",
            "dni": "40358987",
            "eliminado": false
        },
        "empresa": null,
        "eliminado": false
    }
]
```

Endpoint para modificar cliente: POST /api/clientes/{id}

- Ejemplo de body response OK **(status 200)**:

(en la siguiente página)



Al editar un cliente se inserta el id del mismo como variable de ruta y el json con los valores, esto permite poder mutar el tipo del cliente en función del valor de la empresa, es decir un cliente de tipo **PERSONA** puede transformarse en un cliente de tipo **EMPRESA** y viceversa, para que este proceso sea exitoso se llevan a cabo las correspondientes validaciones que impiden que en la edición se relacione un cliente con una empresa o una persona que ya conforman un cliente del tipo que se está editando.

Validaciones de la edición de clientes

Si el json no respeta el formato (se envía de forma errónea) → 400 Bad Request

Si el id de la persona no concuerda con un id de ningún registro persona → 404 Not Found

Si el id de la empresa no concuerda con un id de ningún registro empresa → 404 Not Found

Si ya existe un cliente del tipo persona con los valores de edicion (persona)→ 400 Bad Request

Si ya existe un cliente del tipo empresa con los valores de edicion (empresa)→ 400 Bad Request

Endpoint para borrado lógico de cliente: PATCH /api/clientes/{id}

- Ejemplo de body response OK **(status 202)**:

Validaciones de la eliminación de clientes

Si el id del cliente no corresponde a ningún registro → 404 Not Found

Si el id posee un formato erróneo (si no es un numero) → 400 Bad Request

Puntos de control

- Crear un nuevo Cliente:
    - El body request básico debe contener necesariamente: **persona_id**.
    - Adicionalmente, puede contener ***empresa_id*** que será considerado sólo en caso de tratarse de un cliente *tipo* = "**Empresa**".
    - Existen clientes tipo Empresa y tipo Persona..
    - Se enviará JSON del objeto creado en BBDD con: id, persona, empresa (*null* si es un cliente Persona) y estado de actividad.
    - No se puede crear un cliente que ya exista (contenga misma persona y empresa, o misma persona en caso de ser tipo persona).
- Modificar un Cliente existente:
    - El id proporcionado debe pertenecer a un **Cliente**.
    - l body request básico debe contener necesariamente: **persona_id**.
    - Se modificarán los datos que sean recibidos y válidos.
    - Se enviará JSON del objeto creado en BBDD con: id, persona, empresa (*null* si es un cliente Persona) y estado de actividad.
    - No se puede modificar en un cliente que ya exista (contenga misma persona y empresa, o misma persona en caso de ser tipo persona).
    - No se puede cambiar el nombre de un Producto o Servicio a uno existente.
- Alternar estado de un Cliente (eliminación lógica):
    - El id proporcionado debe pertenecer a un Cliente.
    - Si el id recibido es inválido, se devuelve Status 404.
    - Se enviará JSON del objeto creado en BBDD con: id, persona, empresa (*null* si es un cliente Persona) y estado de actividad.
- Listar un Cliente en detalle:
    - El id proporcionado debe pertenecer a un Cliente.
    - Si el id recibido es inválido, se devuelve Status 404.
    - Se enviará JSON del objeto creado en BBDD con: id, persona, empresa (*null* si es un cliente Persona) y estado de actividad.
- Listar todos los Clientes:
    - No se requiere id ni contenido en la solicitud.
    - Se enviará JSON del array de todos los objetos desde BBDD con: id, persona, empresa (*null* si es un cliente Persona) y estado de actividad. como respuesta exitosa (Status 200).