



# ESPECIFICACIONES TÉCNICAS

Backend de CRUD de Productos/Servicios

Sprint - Deadline	1 - 3/11/23
Jira asociado	SCRUM-26
Responsable	Yamil Leotta
Equipo	<ul style="list-style-type: none"><li>● Miguel Ángel Maisares</li><li>● Lucas Sarfati</li><li>● Javier Huebra</li><li>● Matias Stewart Usher</li><li>● Facundo Oliva</li><li>● Koba Chajchir</li><li>● Yamil Leotta</li></ul>
Versión	1.1

## Contexto

Se realiza un nuevo CRUD respetando la estructura de API y URLs de endpoints utilizada en el CRUD de Usuarios.

## Objetivo

Desarrollar el backend con los endpoints necesarios para realizar el CRUD de Productos/Servicios.

## Alcance

- Crear un nuevo Producto o Servicio utilizando un único endpoint de creación.
- Modificar un Producto o Servicio existente a excepción de su tipo (los Servicios siempre serán Servicios y los Productos siempre serán Productos).
- Eliminar un Producto o Servicio.
- Alternar estado (Eliminación lógica).
- Listar un Producto o Servicio en detalle (con todas sus propiedades). Uso previsto: en modificación.
- Listar todos los Productos y Servicios con sus propiedades principales. Uso previsto: en listado.

## Entregable

- **Endpoint de creación:** **POST** /api/prods\_servs
  - Se devuelve Status 400 si algún dato recibido es inválido.
- **Producto:**
  - Ejemplo de body request correcto:

```
{
  "nombre": "Producto creado correctamente",
  "costo": 123.5,
  "tipo": "PRODUCTO"
}
```

- Ejemplo de body response OK (status 201):

```
{
  "id": 19,
  "nombre": "Producto creado correctamente",
  "tipo": "PRODUCTO",
  "costo": 123.5
}
```

- **Servicio:**

- Ejemplo de body request correcto:

```
{
  "nombre": "Servicio creado correctamente",
  "tipo": "SERVICIO",
  "costo": 1000.5,
  "soporte": 250.75
}
```

- Ejemplo de body response OK (status 201):

```
{
  "id": 20,
  "nombre": "Servicio creado correctamente",
  "tipo": "SERVICIO",
  "costo": 1000.5,
  "soporte": 250.75
}
```

- **Endpoint de modificación:** **POST** /api/prods\_servs/{id}

- Se devuelve Status 404 si el id no existe o es inválido.
- Se devuelve Status 400 si algún dato recibido es inválido.
- Ejemplo de body request correcto:

**POST** /api/prods\_servs/20

```
{
  "nombre": "Modifico el nuevo servicio",
  "costo": 8.5,
  "soporte": 8.75
}
```

- Ejemplo de body response OK (status 200):

```
{
  "id": 20,
  "nombre": "Modifico el nuevo servicio",
  "tipo": "SERVICIO",
  "costo": 8.5,
  "soporte": 8.75
}
```

- **Endpoint de eliminación: DELETE** /api/prods\_servs/{id}
  - No se requiere body request.
  - No se devuelve body response.
  - Se devuelve Status 404 si el id no existe o es inválido
  - Se devuelve Status 204 si el borrado fue exitoso.

- **Endpoint de alternar estado: PATCH** /api/prods\_servs/{id}
  - No se requiere body request.
  - Se devuelve Status 404 si el id no existe o es inválido
  - Se devuelve Status 200 si el borrado fue exitoso.
  - Ejemplo de body response OK (status 200):

```
{
  "id": 3,
  "nombre": "Mantenimiento quincenal",
  "eliminado": true,
  "tipo": "PRODUCTO",
  "costo": 1500.5
}
```

- **Enpoint de listar un Producto o Servicio en detalle: GET** /api/prods\_servs/{id}
  - No se requiere body request.
  - Se devuelve Status 404 si el id no existe o es inválido.
  - Ejemplo de body response OK (status 200):

**GET** /api/prods\_servs/5

```
{
  "id": 5,
  "nombre": "Formateo de PC",
  "eliminado": false,
  "tipo": "SERVICIO",
  "costo": 8500.0,
  "soporte": 500.0
}
```

- **Endpoint de listar todos los Productos y Servicios: GET** /api/prods\_servs/lista
  - No se requiere body request.
  - Ejemplo de body response OK (status 200):

## GET /api/prods\_servs/lista

```
[{"id": 2,
  "nombre": "Modifico el 2 solo el nombre otra vez",
  "eliminado": true,
  "tipo": "SERVICIO",
  "costo": 10.0,
  "soporte": 100.0
},
{"id": 3,
  "nombre": "Teclado",
  "eliminado": true,
  "tipo": "PRODUCTO",
  "costo": 1500.5
},
{"id": 4,
  "nombre": "Mantenimiento bimestral",
  "eliminado": false,
  "tipo": "SERVICIO",
  "costo": 1500.5,
  "soporte": 1250.75
},
{"id": 5,
  "nombre": "Formateo de PC",
```

## Puntos de control

- Crear un nuevo Producto o Servicio:
  - El body request básico debe contener necesariamente: *nombre*, *tipo* y *costo*.
  - Adicionalmente, puede contener *soporte* que será considerado sólo en caso de tratarse de un *tipo* = "SERVICIO".
  - *Tipo* solo puede ser "PRODUCTO" ó "SERVICIO".
  - Se enviará JSON del objeto creado en BBDD con: id, nombre, tipo, costo y soporte (*null* si es un PRODUCTO, o el costo de soporte si es un SERVICIO) como respuesta exitosa (Status 201).
  - No se puede crear un Producto o Servicio con un nombre existente.
- Modificar un Producto o Servicio existente:
  - El id proporcionado debe pertenecer a un Producto o Servicio.
  - El body request básico debe contener opcionalmente: *nombre*, *tipo* y *costo*.
  - Se modificarán los datos que sean recibidos y válidos.
  - Si el id recibido es inválido, se devuelve Status 404.
  - Si algún dato recibido es inválido, se devuelve Status 400.
  - Se enviará JSON del objeto modificado en BBDD con: id, nombre, tipo, costo y soporte (*null* si es un PRODUCTO, o el costo de soporte si es un SERVICIO) como respuesta exitosa (Status 200).
  - No se puede cambiar el nombre de un Producto o Servicio a uno existente.

- Eliminar un Producto o Servicio:
  - El id proporcionado debe pertenecer a un Producto o Servicio.
  - Si el id recibido es inválido, se devuelve Status 404.
  - No enviará contenido en respuesta la exitosa (Status 204).
- Alternar estado de un Producto o Servicio (eliminación lógica):
  - El id proporcionado debe pertenecer a un Producto o Servicio.
  - Si el id recibido es inválido, se devuelve Status 404.
  - Se enviará JSON del objeto creado en BBDD con: id, nombre, tipo, costo y soporte (*null* si es un PRODUCTO, o el costo de soporte si es un SERVICIO) como respuesta exitosa (Status 200).
- Listar un Producto o Servicio en detalle:
  - El id proporcionado debe pertenecer a un Producto o Servicio.
  - Si el id recibido es inválido, se devuelve Status 404.
  - Se enviará JSON del objeto solicitado desde BBDD con: id, nombre, tipo, costo y soporte (*null* si es un PRODUCTO, o el costo de soporte si es un SERVICIO) como respuesta exitosa (Status 200).
- Listar todos los Productos y Servicios:
  - No se requiere id ni contenido en la solicitud.
  - Se enviará JSON del array de todos los objetos desde BBDD con: id, nombre, tipo, costo y soporte (*null* si es un PRODUCTO, o el costo de soporte si es un SERVICIO) para cada Producto o Servicio como respuesta exitosa (Status 200).
  -

## Revisiones

- 1.0. Documento inicial
- 1.1. Modificaciones:
  - Se agrega alternar estado (eliminación lógica).
  - Se cambian los nombres de los campos a español.