

DOCUMENTO DE EVIDENCIAS

Evidencias de Agenda · Ignacio Llorente



Plexus Tech
03-04-2025

Índice

1.0 · GetAll	-----2
2.0 · Filtros	-----3
3.0 · Add (Creación de Employees)	-----6
Endpoint	-----6
Pruebas	-----6
4.0 · Update (Actualización de Employees)	-----10
Endpoint	-----10
Pruebas	-----10
5.0 · Delete (Eliminación de Employees)	----- ¡Error! Marcador no definido.
Endpoint	----- ¡Error! Marcador no definido.
Pruebas	----- ¡Error! Marcador no definido.
6.0 · Casos Masivos	----- ¡Error! Marcador no definido.
Pruebas	----- ¡Error! Marcador no definido.

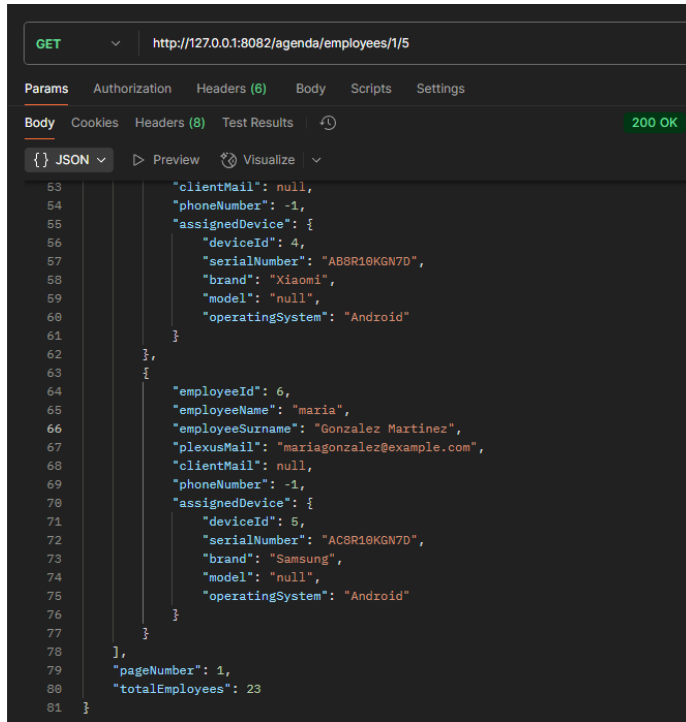
1.0 · GetAll

Endpoint: GET /agenda/employees/{page}/{size} (si no se pone nada el default es 1/10)

Pruebas:

Obtener empleados con paginación:

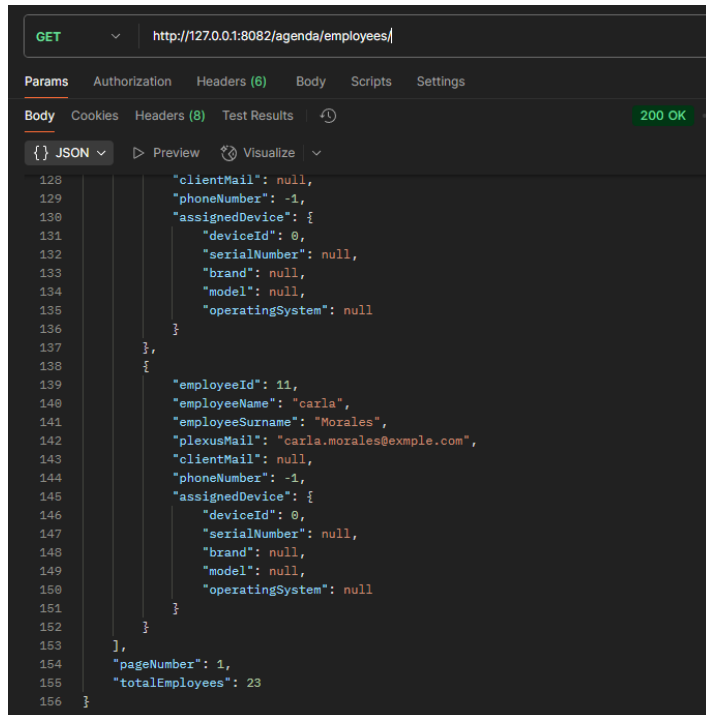
Validar si se obtiene el total de empleados y se devuelve correctamente según los parámetros page y size.



```
GET http://127.0.0.1:8082/agenda/employees/1/5
200 OK
{
  "clientMail": null,
  "phoneNumber": -1,
  "assignedDevice": {
    "deviceId": 4,
    "serialNumber": "AB8R10KGN7D",
    "brand": "Xiaomi",
    "model": "null",
    "operatingSystem": "Android"
  },
  "employeeId": 6,
  "employeeName": "maria",
  "employeeSurname": "Gonzalez Martinez",
  "plexusMail": "mariagonzalez@example.com",
  "clientMail": null,
  "phoneNumber": -1,
  "assignedDevice": {
    "deviceId": 5,
    "serialNumber": "AC8R10KGN7D",
    "brand": "Samsung",
    "model": "null",
    "operatingSystem": "Android"
  },
  "pageNumber": 1,
  "totalEmployees": 23
}
```

Obtener empleados sin page ni size:

Confirmar que los valores por defecto son *page = 1* y *size = 10*.



```
GET http://127.0.0.1:8082/agenda/employees/

200 OK

[{"clientMail": null, "phoneNumber": -1, "assignedDevice": {"deviceId": 0, "serialNumber": null, "brand": null, "model": null, "operatingSystem": null}}, {"employeeId": 11, "employeeName": "carla", "employeeSurname": "Morales", "plexusMail": "carla.morales@example.com", "clientMail": null, "phoneNumber": -1, "assignedDevice": {"deviceId": 0, "serialNumber": null, "brand": null, "model": null, "operatingSystem": null}}, {"pageNumber": 1, "totalEmployees": 23}]
```

Validar asignación de dispositivos:

Para cada empleado, verificar si tiene un dispositivo asignado y manejar correctamente los casos en que no tenga.



```
[{"employeeId": 6, "employeeName": "maria", "employeeSurname": "Gonzalez Martinez", "plexusMail": "mariagonzalez@example.com", "clientMail": null, "phoneNumber": -1, "assignedDevice": {"deviceId": 5, "serialNumber": "AC8R10KGN7D", "brand": "Samsung", "model": null, "operatingSystem": "Android"}}, {"employeeId": 7, "employeeName": "pedro", "employeeSurname": "Jimenez Lopez", "plexusMail": "pedro.jimenez@example.com", "clientMail": null, "phoneNumber": -1, "assignedDevice": {"deviceId": 0, "serialNumber": null, "brand": null, "model": null, "operatingSystem": null}}]
```

(si no tiene dispositivo asignado, sus campos son nulos, si no tiene número de teléfono, será -1)

2.0 · Filtros

Endpoint: GET /agenda/employees/filter/{filterValue}?filterType={name|surname|id}

Pruebas:

Filtrar empleados por nombre (filterType=name):

```
GET http://127.0.0.1:8082/agenda/employees/filter/Nacho?filterType=name&page=1&size=2

Params Authorization Headers (6) Body Scripts Settings
Body Cookies Headers (8) Test Results 200 OK
JSON Preview Visualize
1 {
2   "employees": [
3     {
4       "employeeId": 3,
5       "employeeName": "Nacho",
6       "employeeSurname": "Llorente",
7       "plexusMail": "nacho.llorente@plexus.com",
8       "clientMail": null,
9       "phoneNumber": -1,
10      "assignedDevice": {
11        "deviceId": 1,
12        "serialNumber": "RFBR10KGN7D",
13        "brand": "Samsung",
14        "model": "null",
15        "operatingSystem": "Android"
16      }
17    },
18    {
19      "employeeId": 4,
20      "employeeName": "Nacho",
21      "employeeSurname": "Llorente",
22      "plexusMail": "nacho.llorente2@plexus.com",
23      "clientMail": null,
24      "phoneNumber": -1,
25      "assignedDevice": {
26        "deviceId": 3,
27        "serialNumber": "RFBR10KGN9D",
28        "brand": "Samsung",
29        "model": "null",
30        "operatingSystem": "Android"
31      }
32    }
33  ],
34  "pageNumber": 1,
35  "totalEmployees": 2
36 }
```

Filtrar empleados por apellido (filterType=surname):

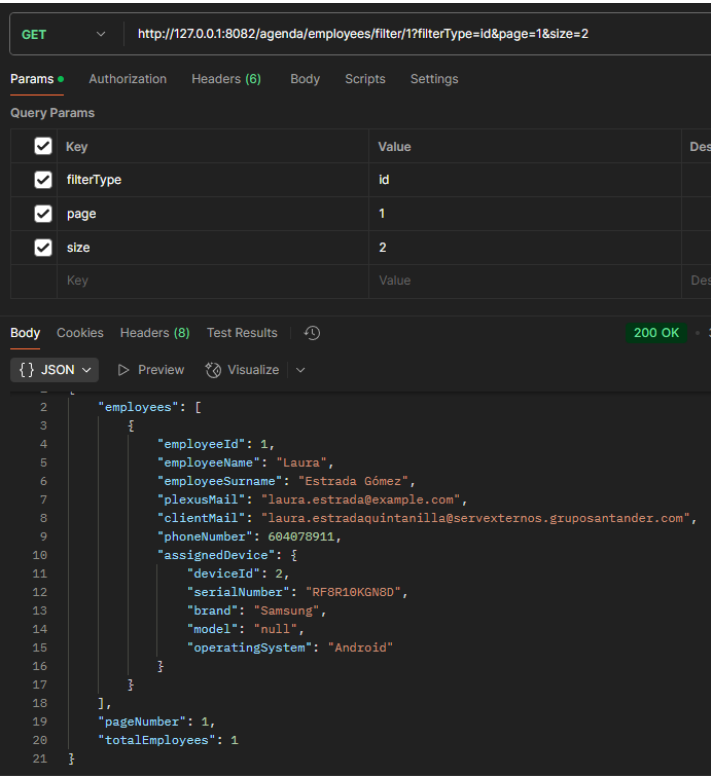
Confirmar que devuelve los resultados esperados.

```
GET http://127.0.0.1:8082/agenda/employees/filter/llo?page=1&size=2&filterType=surname

Params Auth Headers (6) Body Scripts Settings
Body 200 OK 23 ms 837 B
JSON Preview Visualize
5 {
6   "employeeName": "Nacho",
7   "employeeSurname": "Llorente",
8   "plexusMail": "nacho.llorente@plexus.com",
9   "clientMail": null,
10  "phoneNumber": -1,
11  "assignedDevice": {
12    "deviceId": 1,
13    "serialNumber": "RFBR10KGN7D",
14    "brand": "Samsung",
15    "model": "null",
16    "operatingSystem": "Android"
17  }
18 },
19 {
20   "employeeId": 4,
21   "employeeName": "Nacho",
22   "employeeSurname": "Llorente",
23   "plexusMail": "nacho.llorente2@plexus.com",
24   "clientMail": null,
25   "phoneNumber": -1,
26   "assignedDevice": {
27     "deviceId": 3,
28     "serialNumber": "RFBR10KGN9D",
29     "brand": "Samsung",
30     "model": "null",
31     "operatingSystem": "Android"
32   }
33 }
34 ],
35 "pageNumber": 1,
36 "totalEmployees": 2
37 }
```

Filtrar empleados por ID (filterType=id):

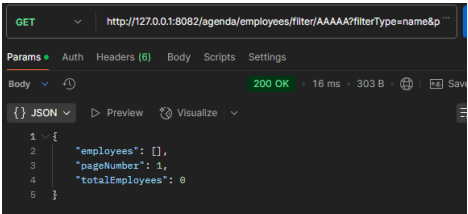
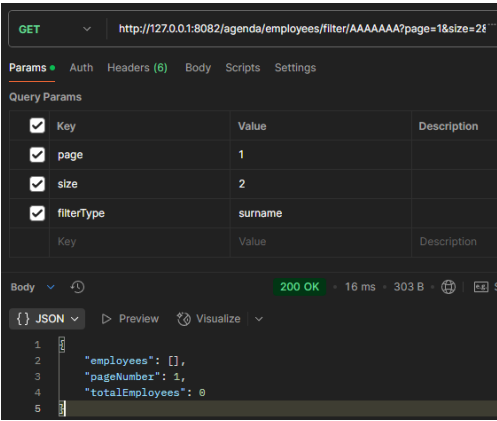
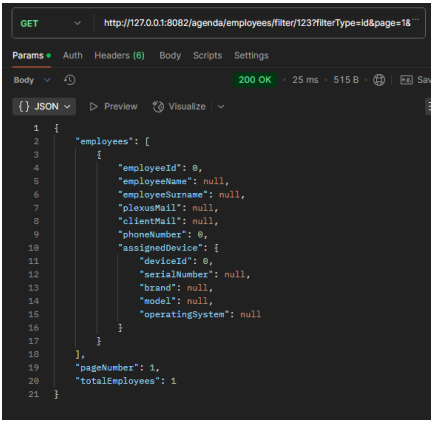
Verificar que devuelve un solo registro con el ID correcto.



Manejo de errores al filtrar con un valor inexistente:

Intentar un filtro con un nombre/apellido/ID que no exista y validar el mensaje de error.

Devuelve un objeto nulo cuando no pones el id bien, y listas vacias cuando no pones nombre o apellidos que existan



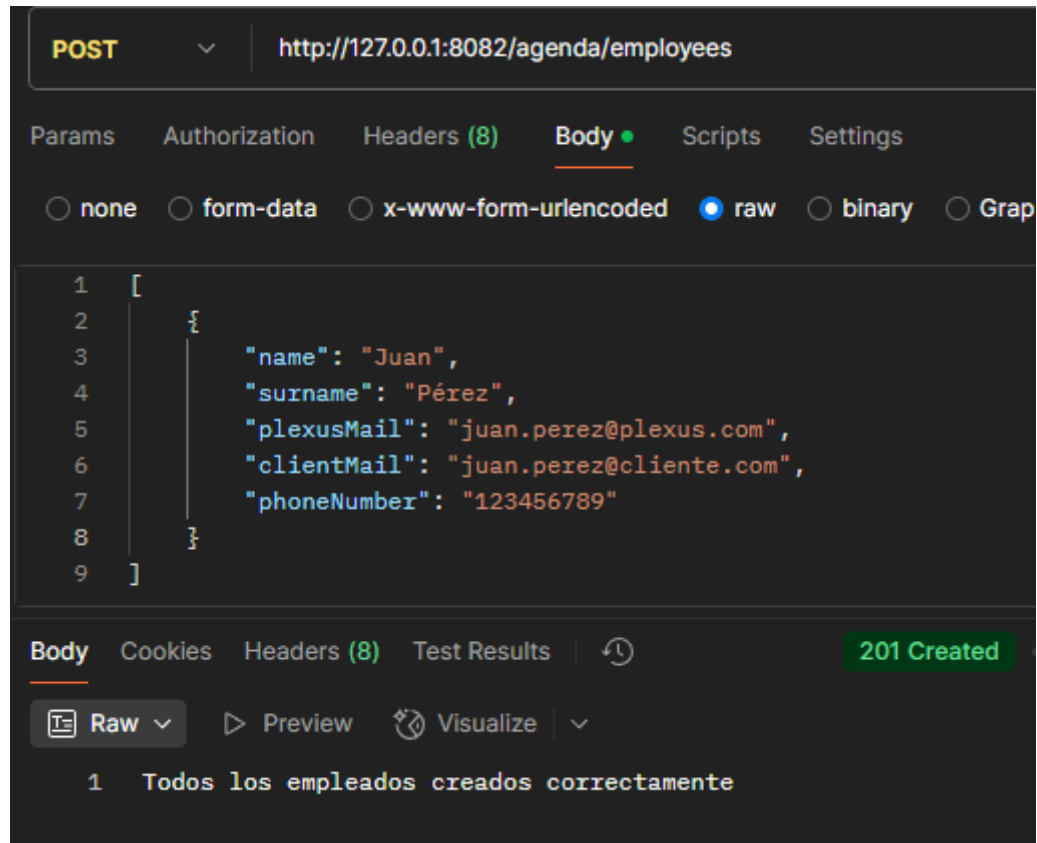
3.0 · Add (Creación de Employees)

Endpoint: POST /agenda/employees

Pruebas:

Crear un employee sin dispositivo asociado:

Debe guardarse sin errores.

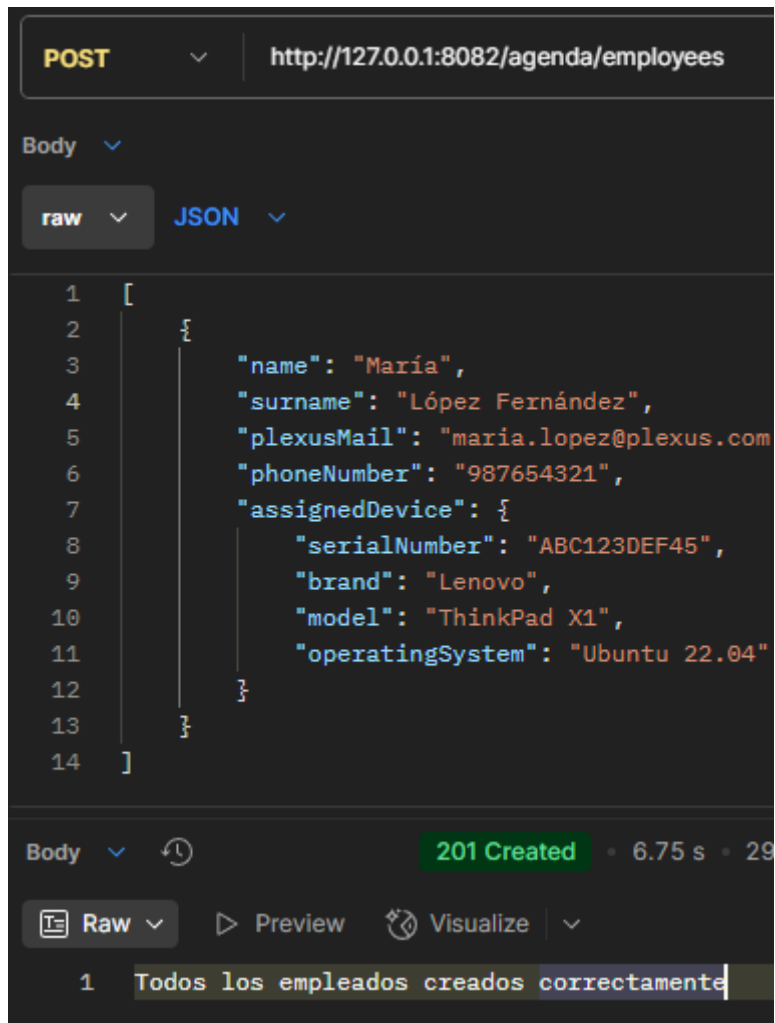


Vemos que se ha agregado:



Crear un employee con un nuevo dispositivo:

Debe crear el dispositivo y asignarlo al empleado.



Vemos que se ha creado el employee con el device



Crear un employee con un dispositivo ya asignado a otro empleado:

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8082/agenda/employees`. The request body is a JSON array containing one object for an employee named Carlos Gómez. The response is a JSON object indicating a constraint error due to a duplicate device serial number.

Request:

```
POST http://127.0.0.1:8082/agenda/employees
```

Body (JSON):

```
[
  {
    "name": "Carlos",
    "surname": "Gómez",
    "plexusMail": "carlos.gomez@plexus.com",
    "clientMail": "c.gomez@empresa.com",
    "assignedDevice": {
      "serialNumber": "ABC123DEF45",
      "brand": "Lenovo",
      "model": "ThinkPad X1",
      "operatingSystem": "Ubuntu 22.04"
    }
  }
]
```

Response (JSON):

```
{
  "details": "Respuesta con múltiples estados",
  "message": "{\\\"Error de constraint\\\":\\\"Hay campos que tienen que ser unicos y no lo son\\\", \\\"Consejo\\\":\\\"verifica los devices que has enviados, comprueba que los datos los has introducido bien\\\", \\\"message\\\":null, \\\"Detalle SQL\\\":\\\"[SQLITE_CONSTRAINT_UNIQUE] A UNIQUE constraint failed (UNIQUE constraint failed: devices.serial_number)\\\", \\\"status\\\":\\\"casi bien\\\"}",
  "status": "casi bien"
}
```

Validaciones del employeee:

- Nombre obligatorio (name)
- Apellido obligatorio (surname)
- Email válido (plexusMail, clientMail)
- Número de teléfono con 9 dígitos (phoneNumber)

Validaciones del dispositivo (assignedDevice):

- serialNumber obligatorio, alfanumérico y con exactamente 11 caracteres
- brand obligatorio, máximo 50 caracteres
- operatingSystem obligatorio, máximo 30 caracteres

Vemos que cumple con las validaciones mostrando un mensaje de error detallado:

The screenshot shows a REST client interface with a POST request to `http://127.0.0.1:8082/agenda/employees`. The request body is a JSON array of three employee objects. The response is a JSON object with a message and a list of errors.

```
1 [
2   {
3     "name": "Maria",
4     "surname": "Garcia López",
5     "plexusMail": "maria.garcia@plexus.es",
6     "clientMail": "maria.garcia@client.com",
7     "phoneNumber": "912345678"
8   },
9   {
10    "name": "Carlos",
11    "surname": "Martinez Ruiz",
12    "plexusMail": "carlos.martinez@plexus.es",
13    "clientMail": "c.martinez@empresa.com",
14    "phoneNumber": "600112233",
15    "assignedDevice": {
16      "serialNumber": "SN789XYZ012",
17      "brand": "Lenovo",
18      "model": "ThinkPad X1 Carbon",
19      "operatingSystem": "Ubuntu 22.04 LTS"
20    }
21  },
22  {
23    "name": "",
24    "surname": "Este apellido es claramente demasiado largo y supera el
25    limite de 150 caracteres que está establecido en las validaciones
26    del modelo de datos",
27    "plexusMail": "correo-invalido",
28    "clientMail": "correo@invalido",
29    "phoneNumber": "ABCDE12345",
30    "assignedDevice": {
31      "serialNumber": "SN_corto",
32      "brand": "",
33      "operatingSystem": "Este sistema operativo es demasiado largo
34      para el campo"
35    }
36  }
37 ]
```

The response body shows a message and a list of errors:

```
1 {
2   "Message": "Se han invalidado 2 de 3 empleados y/o dispositivos.",
3   "Errores": {
4     "Error en el empleado con nombre ": [
5       "El correo debe ser válido",
6       "El phoneNumber debe ser numérico de 9 dígitos",
7       "El nombre es obligatorio"
8     ],
9     "Error en el dispositivo asignado al empleado llamado Este
10    apellido es claramente demasiado largo y supera el limite de
11    150 caracteres que está establecido en las validaciones del
12    modelo de datos": [
13       "tiene que ser alfanumerico y tener exactamente 11 caracteres",
14       "sistema operativo debe tener máximo 50 caracteres",
15       "brand es obligatorio"
16     ]
17   }
18 }
```

4.0 · Update (Actualización de Employees)

Endpoint: PUT /agenda/employees

Pruebas:

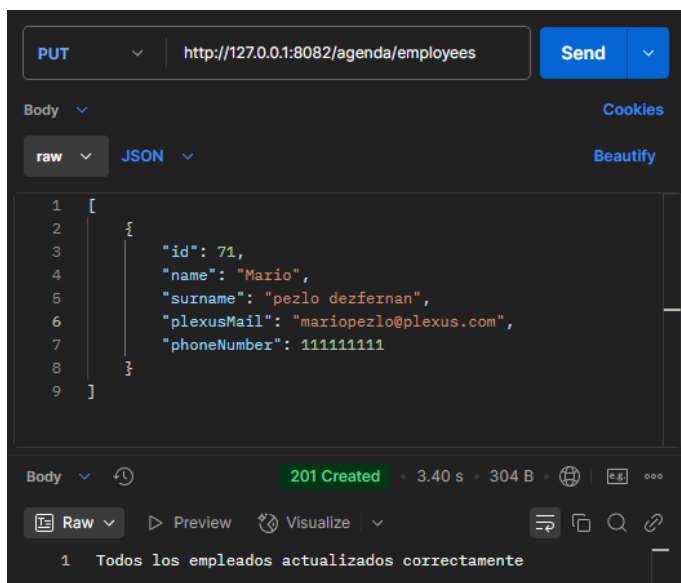
Actualizar datos de un employee existente:

Confirmar que los datos se reflejan correctamente.

Antes:

```
{
  "employeeId": 71,
  "employeeName": "María",
  "employeeSurname": "López Fernández",
  "plexusMail": "maria.lopez@plexus.com",
  "clientMail": null,
  "phoneNumber": 987654321,
```

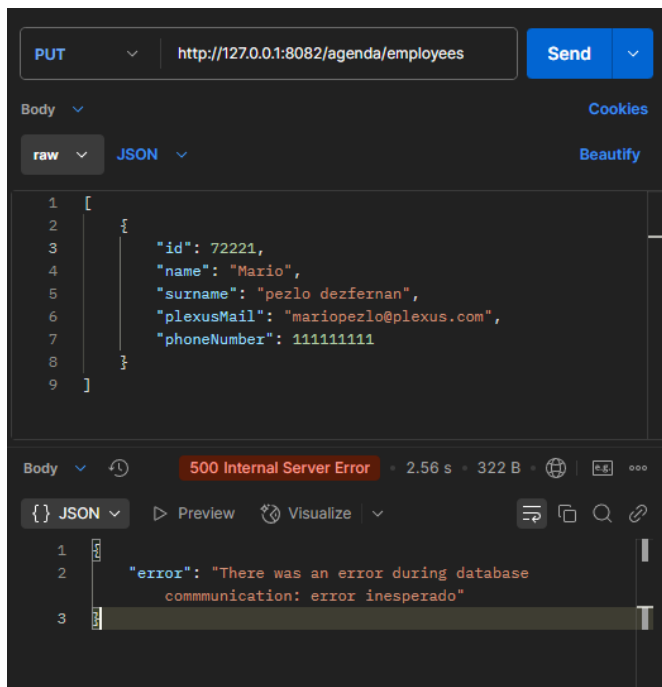
Llamamos:



Vemos que se ha modificado:

```
{
  "id": 71,
  "name": "Mario",
  "surname": "pezlo dezfernan",
  "mailPlexus": "mariopezlo@plexus.com",
  "mailClient": null,
  "clientId": null,
  "phoneNumber": "987654321",
  "phoneSerialNumber": null
}
```

Actualizar un employee inexistente:

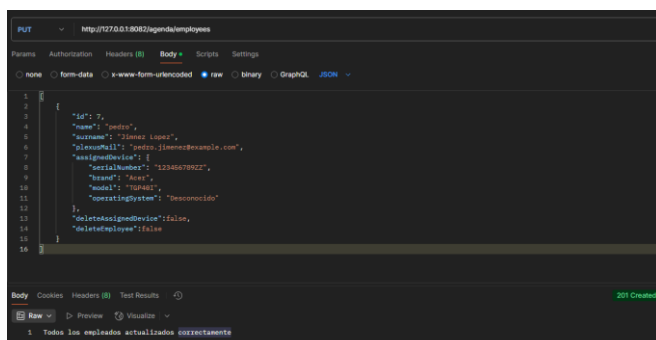


Actualizar un employee con un dispositivo nuevo:

El employee 7 no tiene dispositivo asignado



Le cambiamos el nombre y le metemos un device nuevo

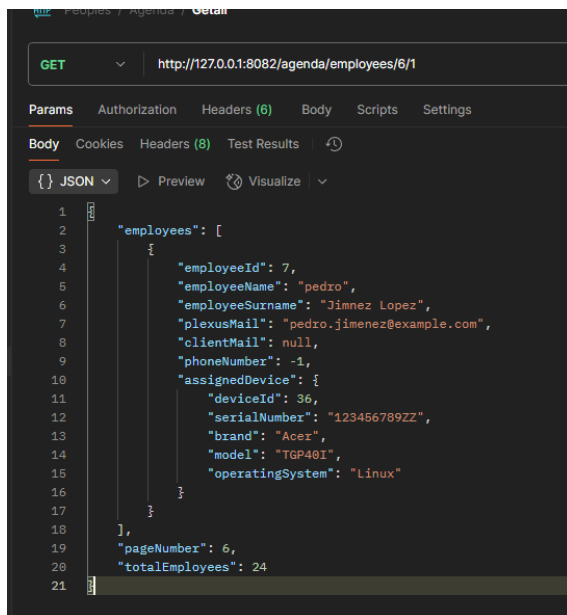


Vemos que se ha actualizado y que se ha creado el device asignado:

```
{
  "employeeId": 7,
  "employeeName": "pedro",
  "employeeSurname": "Jimnez Lopez",
  "plexusMail": "pedro.jimenez@example.com",
  "clientMail": null,
  "phoneNumber": -1,
  "assignedDevice": {
    "deviceId": 36,
    "serialNumber": "123456789ZZ",
    "brand": "Acer",
    "model": "TGP40I",
    "operatingSystem": "Desconocido"
  }
}
```

Modificar un dispositivo existente sin cambiar el empleado:

Vamos a cambiar acer por apple, ahora está así:



```
GET http://127.0.0.1:8082/agenda/employees/6/1

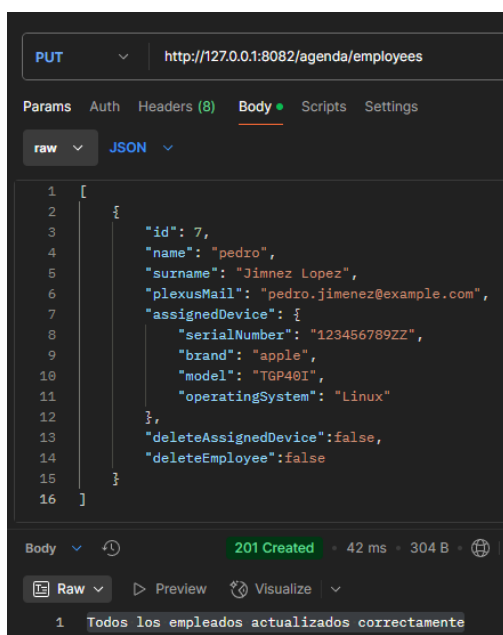
Params
Authorization
Headers (6)
Body
Scripts
Settings

Body
Cookies
Headers (8)
Test Results
Visualize

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

{
  "employees": [
    {
      "employeeId": 7,
      "employeeName": "pedro",
      "employeeSurname": "Jimnez Lopez",
      "plexusMail": "pedro.jimenez@example.com",
      "clientMail": null,
      "phoneNumber": -1,
      "assignedDevice": {
        "deviceId": 36,
        "serialNumber": "123456789ZZ",
        "brand": "Acer",
        "model": "TGP40I",
        "operatingSystem": "Linux"
      }
    }
  ],
  "pageNumber": 6,
  "totalEmployees": 24
}
```

Hacemos la llamada:



```
PUT http://127.0.0.1:8082/agenda/employees

Params
Auth
Headers (8)
Body
Scripts
Settings

raw
JSON

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

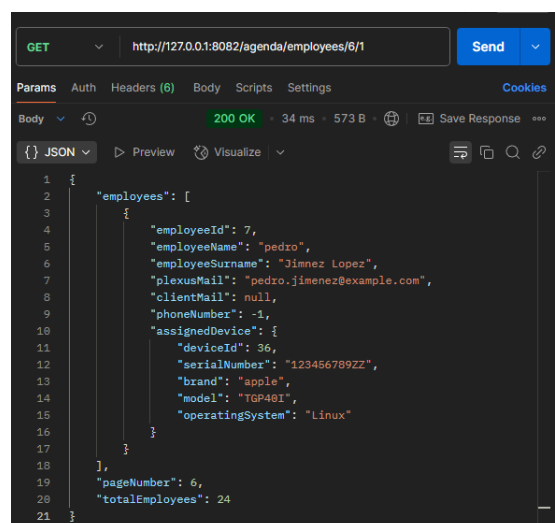
[
  {
    "id": 7,
    "name": "pedro",
    "surname": "Jimnez Lopez",
    "plexusMail": "pedro.jimenez@example.com",
    "assignedDevice": {
      "serialNumber": "123456789ZZ",
      "brand": "apple",
      "model": "TGP40I",
      "operatingSystem": "Linux"
    }
  },
  "deleteAssignedDevice": false,
  "deleteEmployee": false
}
]

Body
201 Created - 42 ms - 304 B

Raw
Preview
Visualize

1 Todos los empleados actualizados correctamente
```

Vemos que se ha actualizado:



```
GET http://127.0.0.1:8082/agenda/employees/6/1

Params
Auth
Headers (6)
Body
Scripts
Settings
Cookies

Body
200 OK - 34 ms - 573 B
Save Response

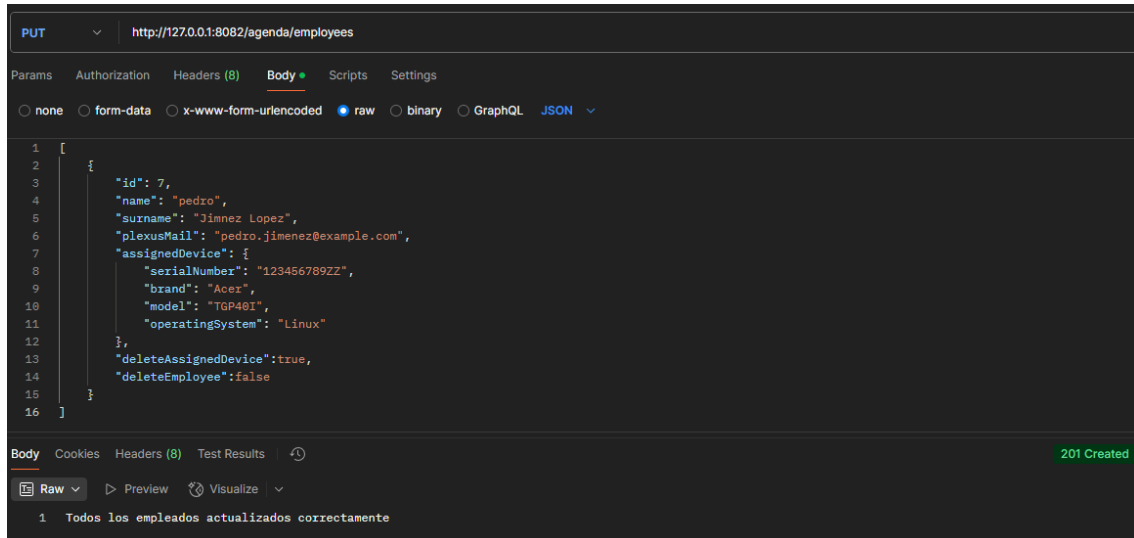
JSON
Preview
Visualize

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

{
  "employees": [
    {
      "employeeId": 7,
      "employeeName": "pedro",
      "employeeSurname": "Jimnez Lopez",
      "plexusMail": "pedro.jimenez@example.com",
      "clientMail": null,
      "phoneNumber": -1,
      "assignedDevice": {
        "deviceId": 36,
        "serialNumber": "123456789ZZ",
        "brand": "apple",
        "model": "TGP40I",
        "operatingSystem": "Linux"
      }
    }
  ],
  "pageNumber": 6,
  "totalEmployees": 24
}
```

Eliminar la asignación del dispositivo:

Si deleteAssignedDevice = true, el dispositivo asignado debe eliminarse del empleado.



Comprobamos que se ha eliminado:

