

# **Manual de usuario**

## **LabTEC-WEB**

# Índice

<b>I.Introducción.....</b>	<b>3</b>
1. Objetivo.....	3
2. Requerimientos.....	3
3. Versión.....	3
<b>II. Uso de página web (Swagger).....</b>	<b>4</b>
<b>IV.Limitaciones.....</b>	<b>6</b>

# I.Introducción

## 1. Objetivo

Esta es la primer versión de LabTec, el cuál tiene como objetivo facilitar la reservación de Activos y laboratorios para profesores y estudiantes, además de facilitar estos también tiene planeado realizar ciertas facilidades a la hora de hacer estos requisitos para que sean más ágiles y rápidos

## 2. Requerimientos

En el caso del servidor API es necesario Node.js el más actualizado para poder utilizar Swagger para realizar las diferentes Request

## 3. Versión

Esta es la primer versión de LabTec

## II. Uso de página web (Swagger)

Para el uso de los Request en Swagger se utiliza de la siguiente manera

En el caso de los GET es simplemente darle al botón de execute, en el caso de la imagen y aparece con Try it out, luego darle al botón grande azul que dice execute y regresará todos los datos que estén en la base de datos

Activos

GET /api/Activos

Parameters

No parameters

Try it out

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5276/api/Activos' \
  -H 'accept: */*'
```

Request URL

http://localhost:5276/api/Activos

Server response

Code Details

200

Response body

```
{
  "placa": "A0001",
  "tipo": "Microscopio",
  "marca": "Nikon",
  "fechaCompra": "2020-01-15",
  "imagenUrl": "http://example.com/microscopio.jpg",
  "ocupado": false
},
{
  "placa": "A0002",
  "tipo": "Proyector",
  "marca": "Epson",
  "fechaCompra": "2019-05-20",
  "imagenUrl": "http://example.com/proyector.jpg",
  "ocupado": false
},
{
  "placa": "A0003",
  "tipo": "Multimetro",
  "marca": "Stern",
  "fechaCompra": "2021-03-30",
  "imagenUrl": "http://example.com/multimetro.jpg",
  "ocupado": false
},
{
  "placa": "A0004",
  "tipo": "Proyector",
  "marca": "Epson",
  "fechaCompra": "2020-01-15",
  "imagenUrl": "http://example.com/proyector.jpg",
  "ocupado": false
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 07 May 2024 05:27:27 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code	Description	Links
200	Success	No links

Para el uso de los POST es necesario rellenar todos los atributos para que se pueda crear una nueva dupla en la base de datos, la misma página le muestra qué tipo de datos tiene

que colocar por lo cual solo debería de seguir la plantilla y colocar los datos y darle a execute para que se realice el request post y se guarde en la base de datos

POST

/api/Activos

Parameters

No parameters

Request body

application/json

```
{
  "placa": "string",
  "tipo": "string",
  "marca": "string",
  "fechaCompra": "string",
  "imagenUrl": "string",
  "cedula": "string",
  "ocupado": true
}
```

Execute

Responses

Code	Description	Links
200	Success	No links

En el caso de los PUTs se tiene que decir que identificador tiene la dupla que se quiere editar los datos, en este caso también muestra una plantilla, pero en este caso no es necesario llenar todos, solo los datos que se quieren editar y funcionaria correctamente

PUT

/api/Activos/{placa}

Parameters

Name	Description
placa * required string (path)	<input type="text" value="placa"/>

Request body

application/json

```
{
  "placa": "string",
  "tipo": "string",
  "marca": "string",
  "fechaCompra": "string",
  "imagenUrl": "string",
  "cedula": "string",
  "ocupado": true
}
```

Execute

Responses

Code	Description	Links
200	Success	No links

## IV.Limitaciones

Las limitaciones más importantes es debido a ciertas restricciones de las foreign key en la base de datos, por lo cual algunos atributos no se pueden enviar si no existen en la base de datos