# **Documentación Api Profesores**

## **Api versión**

Versión 1.0.0

## **API Host**

http://localhost:8080/api/profesores

## **Interacciones con la Api**

## **GET**

<http://localhost:8080/api/profesores>

**Formato de Respuesta:**

Devuelve un array de objetos en formato JSON con todos los profesores.

**Ejemplo:**

[

{

dni: 55000001,

legajo: 1000,

tipoempleado: 'Profesor',

direccion: 'calle 99 567',

telefono: '123456',

email: 'r@yanez',

nombre: 'Roberto',

apellido: 'Yanez'

},

{

dni: 55000004,

legajo: 1500,

tipoempleado: 'Profesor',

direccion: 'Camargo 342',

telefono: '65354353',

email: 'm@perez',

nombre: 'Marcela',

apellido: 'Perez'

}

]

## **GET (Búsqueda por DNI)**

<Http://localhost:8080/api/profesores/?dni=xxxxxxxx>

**Resultado:** Recibe un parámetro con un número de DNI y devuelve un array JSON con el profesor encontrado. Caso de no encontrarlo devuelve un JSON con un error.

**Formato de Respuesta:**

[

{

dni: 9700001,

direccion: 'UNACALLE',

telefono: '666666',

email: 'UN@MAIL',

nombre: 'UNNOMBRE',

apellido: 'UNAPELLIDO',

legajo: 878787,

tipoempleado: 'Profesor'

}

]

**Formato de Respuesta en caso de Error:**

{

error: 400,

msg: 'DNI no encontrado'

}

## **POST**

Http://localhost:8080/api/profesores

Debe recibir un body en formato JSON con el siguiente formato:

{

dni: 9700001,

direccion: 'UNACALLE',

telefono: '666666',

email: 'UN@MAIL',

nombre: 'UNNOMBRE',

apellido: 'UNAPELLIDO',

legajo: 878787,

}

**Resultado:** en caso de inserción correcta devuelve un JSON con el profesor agregado

{

dni: 9700001,

direccion: 'UNACALLE',

telefono: '666666',

email: 'UN@MAIL',

nombre: 'UNNOMBRE',

apellido: 'UNAPELLIDO',

legajo: 878787,

}

**Resultado en caso de error:**

Devuelve un JSON con el error

{

"error": 400,

"msg": "Error en la carga"

}

## **PUT**

<Http://localhost:8080/api/profesores>

**Resultado:** Actualiza los datos de un profesor

Se recibe un parámetro con un body en formato JSON que contiene los datos del profesor a actualizar con el siguiente formato:

{

dni: 9700001,

direccion: 'UNACALLENUEVA',

telefono: '666666',

email: 'UN@MAILNUEVO',

nombre: 'UNNOMBRENUEVO2',

apellido: 'UNAPELLIDONUEVO2',

legajo: 878787,

}

**Formato de Respuesta Ok:**

Devuelve un JSON con el profesor modificado

{

dni: 9700001,

direccion: 'UNACALLENUEVA',

telefono: '666666',

email: 'UN@MAILNUEVO',

nombre: 'UNNOMBRENUEVO2',

apellido: 'UNAPELLIDONUEVO2',

legajo: 878787

}

**Formato de Respuesta cuando hay error:**

{

estado: 400,

descripcion: 'el profesor posee un formato json invalido o faltan datos',

detalles: {

isJoi: true,

name: 'ValidationError',

details: [ [Object] ],

\_object: {

dni: 970000199,

direccion: 'UNACALLENUEVA',

telefono: '666666',

email: 'UN@MAILNUEVO',

nombre: 'UNNOMBRENUEVO2',

apellido: 'UNAPELLIDONUEVO2',

legajo: 878787

}

}

}

## **PUT (Asignar curso a profesor)**

<Http://localhost:8080/api/profesores>/cursoaprofesor/

**Resultado: asigna el curso asignado al profesor.**

Se recibe un body con el formato {curso, legajo}, que indica el id de curso a asignar al profesor con el legajo indicado.

**Resultado Ok:**

Devuelve [ 0 ] si el resultado fue correcto.

**Resultado con errores:**

Devuelve un JSON con el error de la operación:

Si el profesor no existe

{

error: 400,

msg: 'El profesor no existe'

}

O bien si el curso no existe

{

error: 400,

msg: error de curso

}

## **DELETE**

Http://localhost:8080/api/profesores/legajo

Se recibe un parámetro con el número de legajo al final de la ruta. Y elimina el profesor que tenga ese legajo

**Resultado de respuesta Ok:**

En caso de eliminación correcta devuelve el número de celdas afectadas:

“1”

Formato de Respuesta con legajo incorrecto:

{

"error": 400,

"msg": "No se encuentra el empleado"

}