# **Documentación Api Coordinador**

## **Api versión**

Versión 1.0.0

## **API Host**

http://localhost:8080/api/coordinador

## **Interacciones con la Api**

## **GET**

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

**Formato de Respuesta:**

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

**Ejemplo:**

[

{

dni: 55000003,

legajo: 3000,

tipoempleado: 'Coordinador',

direccion: 'Oyola 243',

telefono: '43526525',

email: 'a@alvarez',

nombre: 'Ariadna',

apellido: 'Alvarez'

},

{

dni: 55000007,

legajo: 3500,

tipoempleado: 'Coordinador',

direccion: 'Murillo 666',

telefono: '54334354',

email: 'erne@gonzales',

nombre: 'Ernesto',

apellido: 'Gonzales'

}

]

## **GET (Búsqueda por DNI)**

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

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

**Formato de Respuesta:**

[

{

dni: 55000003,

direccion: 'Oyola 243',

telefono: '43526525',

email: 'a@alvarez',

nombre: 'Ariadna',

apellido: 'Alvarez',

legajo: 3000,

tipoempleado: 'Coordinador'

}

]

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

{

error: 400,

msg: 'DNI no encontrado'

}

## **GET (Búsqueda de Horarios de Profesor)**

<Http://localhost:8080/api/coordinador/getprofesorporlegajo/?legajo=xxxxx>

**Resultado:** Recibe un parámetro con un número de Legajo y devuelve un array JSON con las fechas y horarios encontrados de ese legajo del profesor. Caso de no encontrarlo devuelve un JSON con un error.

**Formato de Respuesta:**

[

{ fechaclase: '2020-06-18T19:18:10.000Z' },

{ fechaclase: '2020-06-22T19:19:00.000Z' },

{ fechaclase: '2020-06-30T19:17:30.000Z' }

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

{

error: 400,

msg: 'El Profesor no tiene cursos activos'

}

## **POST**

Http://localhost:8080/api/coordinador

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 coordinador 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"

}

## **POST (crear nuevo curso como coordinador)**

Http://localhost:8080/api/coordinador/cursonuevo/

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

{

idcurso: 102,

nombrecurso: 'Ingles Tecnico',

fechacurso: '2020-08-03 15:30:00',

}

**Resultado:** Devuelve [ 0 ] si el resultado fue correcto.

**Resultado en caso de error:**

Devuelve un JSON con el error

{

"error": 400,

"msg": "Error en la carga"

}

## **DELETE**

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

Se recibe un parámetro con el número de legajo al final de la ruta. Y elimina el coordinador 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"

}

## **PUT**

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

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

Se recibe un parámetro con un body en formato JSON que contiene los datos del coordinador 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 coordinador modificado

{

dni: 9700001,

direccion: 'UNACALLENUEVA',

telefono: '666666',

email: 'UN@MAILNUEVO',

nombre: 'UNNOMBRENUEVO2',

apellido: 'UNAPELLIDONUEVO2',

legajo: 878787

}

**Formato de Respuesta cuando hay error:**

{

estado: 400,

descripción: 'el coordinador 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 alumno como coordinador)**

<Http://localhost:8080/api/coordinador/cursoalumnocomocoordinador/>

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

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

**Resultado Ok:**

Devuelve [ 0 ] si el resultado fue correcto.

**Resultado con errores:**

Devuelve un JSON con el error de la operación:

Si el alumno no existe

{

error: 400,

msg: 'El alumno no existe'

}

O bien si el curso no existe

{

error: 400,

msg: error de curso

}