

# Arquitectura de Servicios

## U3: Desarrollo de servicios Actividad. - Reporte de funciones AsistenciasREST

MIS. Roberto Suárez Zinzun  
Especialidad: Desarrollador FullStack

Integrantes:

- 21010262** Barriga Garibay Leonardo
- 21010280** García Lira Carlos Humberto
- 21010292** López Quesada Carlos Emmanuel
- 21010310** Ramírez Rodríguez Manuel



# Índice

<b>Evidencias y pruebas de los servicios:</b>	3
<b>Grupos</b>	3
Commits	3
Registrar grupo	4
Consulta general de grupos	4
Consulta por semestre	5
Consultar grupo por ID	6
Actualizar grupo	7
Eliminar grupo	8
Comprobación MongoDB	8
<b>Actividades</b>	9
Commits	9
Registrar actividad	10
Consultar general de actividades	11
Consultar de actividad por Id	12
Ejemplo de error	12
Actualizar actividad	13
Eliminar actividad	14
Comprobación MongoDB	14
<b>Usuarios</b>	15
Commits	15
Registrar usuario	16
Consultar general de usuarios	25
Consultar de actividad por Id	26
Actualizar usuario (ejemplo con coordinador)	27
Eliminar usuario	28
Comprobación MongoDB	29
<b>Ciclos</b>	30
Commits	30
Registrar ciclos	30
Consulta general de ciclos	31
Consulta un ciclo por su id	32

Actualizar Ciclo .....	33
Eliminar Ciclo .....	34
Comprobación MongoDB.....	35
<b>Carreras</b> .....	36
Registrar carrera .....	36
Consulta general de carreras .....	37
Consulta un carrera por su id .....	37
Actualizar Carrera.....	38
Eliminar Carrera .....	39
Comprobación MongoDB.....	40

## Evidencias y pruebas de los servicios:

### Grupos

### Commits

Se realizaron los siguientes commits por Carlos Humberto García Lira en Git referente al servicio de grupos, que se puede resumir en las siguientes rutas, operaciones, salidas.

The first screenshot shows a list of files on the left and a code editor on the right. The files list includes 'Modulo de grupos completo', 'Grupos DAO, CRUD, Router', 'Avance Actividades', 'Gitignore', 'InstalacionesParaFuncionalidades', 'REG\_USUARIO', 'Operación prueba Reg\_User', and 'Primeros pasos actividades'. The code editor shows the 'main.py' file with the following code:

```
@@ -1,13 +1,14 @@
1 1 import uvicorn
2 2 from fastapi import FastAPI
3 3 from dao.database import Conexion
4 4 - from routers import usuariosRouter, actividadesRouter
4 4 + from routers import usuariosRouter, actividadesRouter, gruposRouter
5 5
6 6 app = FastAPI()
7 7
8 8 # Incluir routers
9 9 app.include_router(usuariosRouter.router)
10 10 app.include_router(actividadesRouter.router)
11 11 + app.include_router(gruposRouter.router)
12 12
13 13 @app.get("/")
14 14 async def root():
```

The second screenshot shows the same files list and code editor. The code editor now shows the 'dao/gruposDAO.py' file with the following code:

```
@@ -1,4 +1,4 @@
1 1 - from models.gruposModel import GrupoInsert, Salida, GrupoSelect, GruposSalida, GrupoSalida
1 1 + from models.gruposModel import GrupoInsert, GrupoUpdate, Salida, GrupoSelect, GruposSalida, GrupoSalida
2 2 from fastapi.encoders import jsonable_encoder
3 3 from bson import ObjectId
4 4
5 5
6 6 @@ -6,6 +6,9 @@ class GrupoDAO:
6 6 def __init__(self, db):
7 7 self.db = db
```

### Rutas del servicio en swagger:

Grupos				
GET	/grupos/	Consultar todos los grupos		
POST	/grupos/	Crear un nuevo grupo		
GET	/grupos/semestre/{semestre}	Consultar grupos por semestre		
GET	/grupos/{idGrupo}	Consultar un grupo por su ID		
PUT	/grupos/{idGrupo}	Actualizar un grupo		
DELETE	/grupos/{idGrupo}	Eliminar un grupo		

## Registrar grupo

**POST** /grupos/ Crear un nuevo grupo

Parameters Try it out

No parameters

Request body required application/json

Example Value | Schema

```
{
  "nombre": "string",
  "semestre": 0,
  "ciclo": "string",
  "carrera": 0,
  "tutor": "string",
  "alumnos": [
    "string"
  ]
}
```

Respuesta de agregado correctamente:

201

Response body

```
{
  "estatus": "OK",
  "mensaje": "Grupo creado con ID: 682433dbe6e6e5c5d49915a4",
  "grupo": {
    "id": "682433dbe6e6e5c5d49915a4",
    "nombre": "A",
    "semestre": 0,
    "ciclo": "67d5069a5c4a73aca45548b8",
    "carrera": 1,
    "tutor": "6823a1a6f9304440ce512619",
    "alumnos": [
      "6823a125f9304440ce512618"
    ]
  }
}
```

Download

Ejemplo de error:

Code | Details

201

Response body

```
{
  "estatus": "ERROR",
  "mensaje": "Uno o más alumnos especificados no existen o no son de tipo alumno",
  "grupo": null
}
```

Download

## Consulta general de grupos

**GET** /grupos/ Consultar todos los grupos

Parameters Try it out

No parameters

Respuesta correcta:

Code

Details

200

Response body

```
{
  "estatus": "OK",
  "mensaje": "Listado de grupos.",
  "grupos": [
    {
      "id": "67d507465c4a73aca45548c7",
      "nombre": "4a",
      "semestre": 4,
      "ciclo": "67d5069a5c4a73aca45548b8",
      "carrera": 1,
      "tutor": "681d7507e0cf7d63ec962",
      "alumnos": [
        "68226ebfdb6c2962590cc128"
      ]
    },
    {
      "id": "67d507465c4a73aca45548c8",
      "nombre": "6B",
      "semestre": 6,
      "ciclo": "67d5069a5c4a73aca45548b8",
      "carrera": 1,
      "tutor": "681d7507e0cf7d63ec962",
      "alumnos": [
        "68226ebfdb6c2962590cc128"
      ]
    }
  ],
  "id": "682433dbe6e6e5c5d49915a4"
}
```

Download

Ejemplo de error:

Code

Details

200

Response body

```
{
  "estatus": "ERROR",
  "mensaje": "Error al consultar los grupos, consulte al administrador.",
  "grupos": []
}
```

Download

## Consulta por semestre

GET

/grupos/semestre/{semestre}

Consultar grupos por semestre

⬆

Parameters

Try it out

Name	Description
<b>semestre</b> * required	
integer	
(path)	<input type="text" value="semestre"/>

Respuesta correcta:

200

Response body

```
{
  "estatus": "OK",
  "mensaje": "Grupos del semestre 8.",
  "grupos": [
    {
      "id": "682433dbe6e6e5c5d49915a4",
      "nombre": "A",
      "semestre": 8,
      "ciclo": "67d5069a5c4a73aca45548b8",
      "carrera": 1,
      "tutor": "6823a1a6f9304440ce512619",
      "alumnos": [
        "6823a125f9304440ce512618"
      ]
    }
  ]
}
```

Download


Ejemplo de error:

400  
*Undocumented*

Error: Bad Request

Response body

```
{
  "detail": "El semestre debe estar entre 1 y 9"
}
```

 Download

## Consultar grupo por ID

GET

/grupos/{idGrupo} Consultar un grupo por su ID

^

Parameters

Try it out


Name	Description
<b>idGrupo</b> * required	
string	idGrupo
(path)	

Respuesta de búsqueda correcta:

200

Response body

```
{
  "estatus": "OK",
  "mensaje": "Grupo encontrado.",
  "grupo": {
    "id": "682433dbe6e6e5c5d49915a4",
    "nombre": "A",
    "semestre": 8,
    "ciclo": "67d5069a5c4a73aca45548b8",
    "carrera": 1,
    "tutor": "6823a1a6f9304440ce512619",
    "alumnos": [
      "6823a125f9304440ce512618"
    ]
  }
}
```

 Download

Ejemplo de error:

200

Response body

```
{
  "estatus": "ERROR",
  "mensaje": "Grupo con ID 682433dbe6e6e5c5d49915a2 no encontrado.",
  "grupo": null
}
```

 Download

Response headers

## Actualizar grupo

PUT

/grupos/{idGrupo}

Actualizar un grupo

⬆

Parameters

Try it out

Name	Description
<b>idGrupo</b> <small>required</small>	
string	idGrupo
(path)	

Request body required

application/json

Example Value | Schema

```
{
  "nombre": "string",
  "semestre": 0,
  "ciclo": "string",
  "carrera": 0,
  "tutor": "string",
  "alumnos": [
    "string"
  ]
}
```

Respuesta actualizado correctamente:

200

Response body

```
{
  "estatus": "OK",
  "mensaje": "Grupo actualizado correctamente.",
  "grupo": {
    "id": "682433dbe6e6e5c5d49915a4",
    "nombre": "B",
    "semestre": 8,
    "ciclo": "67d5069a5c4a73aca45548b8",
    "carrera": 1,
    "tutor": "6823a1a6f9304440ce512619",
    "alumnos": [
      "6823a125f9304440ce512618"
    ]
  }
}
```

Download

Ejemplo de error:

200

Response body

```
{
  "estatus": "ERROR",
  "mensaje": "El tutor especificado no existe o no es de tipo tutor",
  "grupo": null
}
```

Download



## Eliminar grupo

**DELETE** /grupos/{idGrupo} Eliminar un grupo

Parameters

Cancel

Name	Description
<b>idGrupo</b> * required	
string	idGrupo
(path)	

Respuesta de eliminado correctamente:

200

Response body

```
{  "estatus": "OK",  "mensaje": "Grupo eliminado exitosamente. ID: 682433dbe6e6e5c5d49915a4"}
```

Download

Ejemplo de error:

200

Response body

```
{  "estatus": "ERROR",  "mensaje": "Grupo con ID 682433dbe6e6e5c5d49915a2 no encontrado."}
```

Download

## Comprobación MongoDB

**Compass**

My Queries

CONNECTIONS (2)

Search connections

159.54.150.147:27017

AsistenciasFastAPI

actividades

carreras

ciclos

**grupos**

ubicaciones

usuarios

viewUsuariosGeneral

viewUsuariosID

admin

config

default

local

grupos usuarios ciclos +

159.54.150.147:27017 > AsistenciasFastAPI > grupos

Documents 0 Aggregations Schema Indexes 1 Validation

+

ADD DATA EXPORT DATA UPDATE DELETE

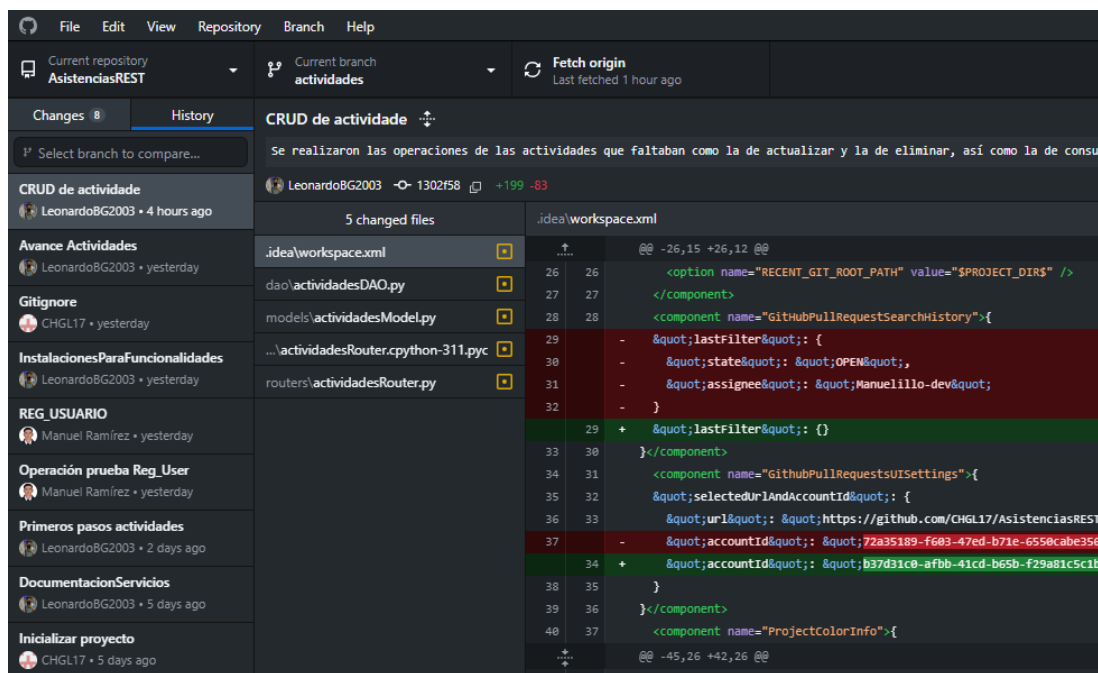
```
_id: ObjectId('6824370bf39dca874a830108')
nombre: "A"
semestre: 8
ciclo: "67d5069a5c4a73aca45548b8"
carrera: 1
tutor: "6823a1a6f9304440ce512619"
alumnos: Array (1)
```

```
_id: ObjectId('68243722f39dca874a830109')
nombre: "6"
semestre: 7
ciclo: "67d5069a5c4a73aca45548b8"
carrera: 1
tutor: "6823a1a6f9304440ce512619"
alumnos: Array (1)
```

## Actividades

### Commits

Se realizaron los siguientes commits en Git realizados Leonardo Barriga Garibay referente al servicio de actividades, que se puede resumir en las siguientes rutas, operaciones, salidas.



### Rutas del servicio en swagger:

#### Actividades

GET	/actividades/	Consultar todas las actividades	^
POST	/actividades/	Crear una nueva actividad	
GET	/actividades/{idActividad}	Consultar una actividad por su ID	
PUT	/actividades/{idActividad}	Actualizar una actividad	
DELETE	/actividades/{idActividad}	Eliminar una actividad	

## Registrar actividad

**POST** /actividades/ Crear una nueva actividad

Parameters Try it out

No parameters

Request body required application/json

Example Value | Schema

```
{
  "nombre": "string",
  "descripcion": "string",
  "estatus": "Por realizar",
  "obligatoria": false
}
```

Respuesta de agregado correctamente:

Code Details

200

Response body

```
{
  "estatus": "OK",
  "mensaje": "Actividad creada exitosamente"
}
```

Response headers

```
content-length: 58
content-type: application/json
date: Wed, 14 May 2025 06:51:25 GMT
server: uvicorn
```

Ejemplo de error

Code Details

422

Error: Unprocessable Entity

Response body

```
{
  "detail": [
    {
      "type": "literal_error",
      "loc": [
        "body",
        "estatus"
      ],
      "msg": "Input should be 'Por realizar', 'Cancelada' or 'Realizada'",
      "input": "Por",
      "ctx": {
        "expected": "'Por realizar', 'Cancelada' or 'Realizada'"
      }
    }
  ]
}
```

## Consultar general de actividades

**Actividades**

GET /actividades/ Consultar todas las actividades

Parameters

No parameters

Execute

Clear

Cancel

Respuesta correcta:

200

Response body

```
{
  "estatus": "OK",
  "mensaje": "Listado de actividades.",
  "actividades": [
    {
      "nombre": "3er Congreso de las Ciencias de la Computación",
      "descripcion": "Congreso anual de las ciencias de la computación",
      "estatus": "Por realizar",
      "obligatoria": true
    },
    {
      "nombre": "Taller de programación web",
      "descripcion": "Introducción a HTML, CSS y JavaScript",
      "estatus": "Por realizar",
      "obligatoria": false
    },
    {
      "nombre": "Examen de medio curso",
      "descripcion": "Evaluación de los temas vistos en la primera mitad del semestre",
      "estatus": "Por realizar",
      "obligatoria": true
    },
    {
      "nombre": "Presentación de proyectos finales",
      "descripcion": "Exposición de los proyectos desarrollados por los alumnos",
      "estatus": "Por realizar",
      "obligatoria": true
    }
  ]
}
```

Download

Response headers

```
content-length: 1031
content-type: application/json
date: Wed, 14 May 2025 07:29:12 GMT
server: unicorn
```

## Consultar de actividad por Id

**GET** /actividades/{idActividad} Consultar una actividad por su ID

Parameters

Try it out

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

Respuesta correcta:

**idActividad** \* required  
string  
(path)

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/actividades/68243d6d0146a290166a246f' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/actividades/68243d6d0146a290166a246f
```

Server response

Code	Details
200	<div>Response body</div> <div><pre>{   "estatus": "OK",   "mensaje": "Actividad encontrada.",   "actividad": {     "nombre": "Platica de residencias",     "descripcion": "Platica de residencias impartida en el edificio de industrial",     "estatus": "Por realizar",     "obligatoria": true   } }</pre></div> <div>Download</div>

Ejemplo de error

**idActividad** \* required  
string  
(path)

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/actividades/68243d6d0146a290166a46f' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/actividades/68243d6d0146a290166a46f
```

Server response

Code	Details
200	<div>Response body</div> <div><pre>{   "estatus": "ERROR",   "mensaje": "Error al consultar la actividad, consulte al administrador.",   "actividad": null }</pre></div> <div>Download</div>

## Actualizar actividad

**PUT** /actividades/{idActividad} Actualizar una actividad

Parameters

Name

Description

**idActividad** required

string  
(path)

68243d6d0146a290166a246f

Cancel

Reset

## Respuesta correcta

**Code** Details

200

Response body

```
{
  "estatus": "OK",
  "mensaje": "Actividad actualizada correctamente.",
  "actividad": {
    "nombre": "Platica de residencias",
    "descripcion": "Platica de residencias impartida en el edificio de industrial",
    "estatus": "Realizada",
    "obligatoria": false
  }
}
```

Response headers

```
content-length: 233
content-type: application/json
date: Wed, 14 May 2025 07:35:53 GMT
server: uvicorn
```

Download

## Ejemplo de error

**Code** Details

422

Error: Unprocessable Entity

Response body

```
{
  "detail": [
    {
      "type": "json_invalid",
      "loc": [
        "body",
        105
      ],
      "msg": "JSON decode error",
      "input": {},
      "ctx": {
        "error": "Expecting value"
      }
    }
  ]
}
```

Response headers

```
content-length: 126
content-type: application/json
date: Wed, 14 May 2025 07:37:04 GMT
server: uvicorn
```

Download

## Eliminar actividad

**DELETE** /actividades/{idActividad} Eliminar una actividad

Parameters

Cancel

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

Execute

### Respuesta correcta

Code Details

200

Response body

```
{
  "estatus": "OK",
  "mensaje": "Actividad eliminada exitosamente. ID: 68243d6d0146a290166a246f"
}
```

Response headers

```
content-length: 91
content-type: application/json
date: Wed, 14 May 2025 07:39:35 GMT
server: uvicorn
```

### Ejemplo de error

Code Details

200

Response body

```
{
  "estatus": "ERROR",
  "mensaje": "Error interno al eliminar la actividad."
}
```

Response headers

```
content-length: 71
content-type: application/json
date: Wed, 14 May 2025 07:40:00 GMT
server: uvicorn
```

## Comprobación MongoDB

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (2)

Search connections

159.54.150.147:27017

AsistenciasFastAPI

actividades

carreras

ciclos

grupos

ubicaciones

usuarios

viewUsuariosGeneral

viewUsuariosID

admin

actividades

159.54.150.147:27017 > AsistenciasFastAPI > actividades

Documents 2 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

ADD DATA EXPORT DATA UPDATE DELETE

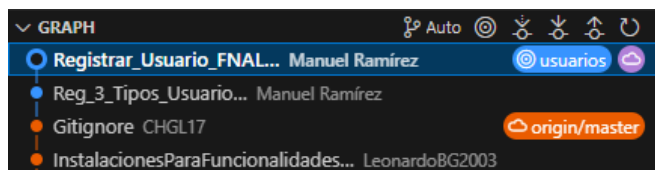
```
_id: ObjectId('6824494b57744fe64f2f6a58')
nombre: "Concurso de programacion"
descripcion: "Concurso interno de programacion"
estatus: "Cancelada"
obligatoria: true
```

```
_id: ObjectId('6824498c57744fe64f2f6a59')
nombre: "Platica de residencias"
descripcion: "Platica de residencias en el edificio de industrial"
estatus: "Por realizar"
obligatoria: true
```

## Usuarios

## Commits

Desde el Visual Studio Code por Manuel Ramírez Rodríguez:



Desde Git-Hub Desktop:

Current repository	Current branch	Fetch origin
AsistenciasREST	usuarios	Last fetched just now
Changes	History	Showing changes from 5 commits
Select branch to compare...		12 changed files
		.idea\workspace.xml
Modificar_Usuario_FINAL Manuel Ramírez • 10 hours ago		↑ 5 5 @@ -5
Eliminar_Usuario_FINAL Manuel Ramírez • 13 hours ago		6 6 </co
Consulta_Usuario_Gral Manuel Ramírez • 13 hours ago		7 7 <
Consulta_Usuario_ID Manuel Ramírez • 14 hours ago		8 8 -
Registrar_Usuario_FINAL Manuel Ramírez • 16 hours ago		9 9 -
Reg_3_Tipos_Usuario Manuel Ramírez • 22 hours ago		10 10 -
Gitignore CHGL17 • yesterday		11 8
InstalacionesParaFuncionalidades LeonardoBG2003 • yesterday		12 9
REG_USUARIO Manuel Ramírez • 2 days ago		13 10
Operación prueba Reg_User Manuel Ramírez • 2 days ago		↓ 39 36 @@ -5
Primeros pasos actividades LeonardoBG2003 • 2 days ago		40 37 &
DocumentacionServicios LeonardoBG2003 • 5 days ago		41 38 }</co
Inicializar proyecto CHGL17 • 5 days ago		39 39 + <co
		40 40 + <
		41 41 + </C
		42 42 <co
		43 43 &qu
		44 44 }</co
		↑ 54 54 &
		55 55 &
		56 56 &
		57 57 - &
		57 57 + &
		58 58 &
		59 59 &
		60 60 &
		↑ @@ -1



## Rutas del swagger:



### Usuarios

POST	/usuarios/publico/alumno	Registro público para alumnos	▼
POST	/usuarios/privado/tutor	Registro para tutores	▼
POST	/usuarios/privado/coordinador	Registro para coordinadores	▼
GET	/usuarios/{id_usuario}	Consultar un usuario por su ID	▼
DELETE	/usuarios/{id_usuario}	Eliminar un usuario permanentemente	▼
GET	/usuarios/general/	Consulta general de usuarios	▼
PUT	/usuarios/alumno/{id_usuario}	Actualizar información de un alumno	▼
PUT	/usuarios/tutor/{id_usuario}	Actualizar información de un tutor	▼
PUT	/usuarios/coordinador/{id_usuario}	Actualizar información de un coordinador	▼

## Registrar usuario

1. Primeras validaciones: Error, nombre o apellidos muy corto (ejemplo con alumno).  
Lo siguiente funciona igual con todos los tipos de usuario (con sus respectivas entradas):

### Usuarios

<b>POST</b> /usuarios/publico/alumno Registro público para alumnos	<b>Usuarios</b>
<b>Parameters</b>	<b>Parameters</b>
No parameters	No parameters
<b>Request body</b> required	<b>Request body</b> required
<pre>{   "email": "alumno1@example.com",   "nombre": "J",   "apellidos": "Pérez López",   "tipo": "alumno",   "password": "Pass1234!",   "alumno": {     "noControl": "210000002",     "semestre": 3,     "carrera": 1   },   "fechaRegistro": "2025-05-13T18:13:17.154Z" }</pre>	<pre>{   "email": "alumno1@example.com",   "nombre": "Juan",   "apellidos": "P",   "tipo": "alumno",   "password": "Pass1234!",   "alumno": {     "noControl": "210000002",     "semestre": 3,     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:07:18.997Z" }</pre>

## Salidas:

<b>Request URL</b>	<b>Request URL</b>
<code>http://127.0.0.1:8000/usuarios/publico/alumno</code>	<code>http://127.0.0.1:8000/usuarios/publico/alumno</code>
<b>Server response</b>	<b>Server response</b>
<b>Code</b> <b>Details</b>	<b>Code</b> <b>Details</b>
400	400
Error: Bad Request	Error: Bad Request
<b>Response body</b>	<b>Response body</b>
<pre>{   "detail": "El nombre debe tener al menos 2 caracteres" }</pre>	<pre>{   "detail": "Los apellidos deben tener al menos 2 caracteres" }</pre>

Errores, nombres con números o sin espacios invalidados.

## Usuarios

**POST** /usuarios/publico/alumno Registro público para alumnos

**Parameters**

No parameters

**Request body** *required*

```
{
  "email": "alumno1@example.com",
  "nombre": "Juan2",
  "apellidos": "Pérez",
  "tipo": "alumno",
  "password": "Pass1234!",
  "alumno": {
    "noControl": "21000002",
    "semestre": 3,
    "carrera": 1
  },
  "fechaRegistro": "2025-05-13T19:07:18.997Z"
}
```

## Usuarios

**POST** /usuarios/publico/alumno Registro público para alumnos

**Parameters**

No parameters

**Request body** *required*

```
{
  "email": "alumno1@example.com",
  "nombre": "Juan",
  "apellidos": "Pérez2",
  "tipo": "alumno",
  "password": "Pass1234!",
  "alumno": {
    "noControl": "21000002",
    "semestre": 3,
    "carrera": 1
  },
  "fechaRegistro": "2025-05-13T19:07:18.997Z"
}
```

## Salidas:

Request URL		Request URL	
http://127.0.0.1:8000/usuarios/publico/alumno		http://127.0.0.1:8000/usuarios/publico/alumno	
Server response		Server response	
Code	Details	Code	Details
400	Error: Bad Request	400	Error: Bad Request
Response body		Response body	
{ "detail": "El nombre solo puede contener letras y espacios" }		{ "detail": "Los apellidos solo pueden contener letras y espacios" }	

- Segundas validaciones: Error, número de caracteres inválidos y al menos una mayúscula (ejemplo con tutor).

**POST** /usuarios/privado/tutor Registro para tutores (requiere autenticación)

**Parameters**

No parameters

**Request body** *required*

```
{
  "email": "tutoruser@example.com",
  "nombre": "Aron",
  "apellidos": "Rocha",
  "tipo": "tutor",
  "password": "A9",
  "tutor": {
    "noDocente": "A534458816",
    "horasTutoria": 12,
    "carrera": 1
  },
  "fechaRegistro": "2025-05-13T19:14:41.911Z"
}
```

**Request URL**

http://127.0.0.1:8000/usuarios/privado/tutor

**Server response**

Code	Details
400	Error: Bad Request
Response body	
{ "detail": "La contraseña debe tener al menos 8 caracteres" }	

**POST** /usuarios/privado/tutor Registro para tutores (requiere autenticación)

**Parameters**

No parameters

**Request body** *required*

```
{
  "email": "tutoruser@example.com",
  "nombre": "Aron",
  "apellidos": "Rocha",
  "tipo": "tutor",
  "password": "a9102548",
  "tutor": {
    "noDocente": "A534458816",
    "horasTutoria": 12,
    "carrera": 1
  },
  "fechaRegistro": "2025-05-13T19:14:41.911Z"
}
```

## Salidas:

**Request URL**

http://127.0.0.1:8000/usuarios/privado/tutor

**Server response**

Code	Details
400	Error: Bad Request
Response body	
{ "detail": "La contraseña debe contener al menos una mayúscula" }	

Errores, La contraseña debe contener al menos un número y al menos un carácter especial.

POST /usuarios/privado/tutor Registro para tutores (requiere autenticación)	POST /usuarios/privado/tutor Registro para tutores (requiere autenticación)
Parameters	Parameters
No parameters	No parameters
Request body <small>required</small>	Request body <small>required</small>
<pre>{  "email": "tutoruser@example.com",  "nombre": "Aron",  "apellidos": "Rocha",  "tipo": "tutor",  "password": "A*ASFTGJKLL",  "tutor": {    "noDocente": "AS34458816",    "horasTutoria": 12,    "carrera": 1  },  "fechaRegistro": "2025-05-13T19:14:41.911Z"}</pre>	<pre>{  "email": "tutoruser@example.com",  "nombre": "Aron",  "apellidos": "Rocha",  "tipo": "tutor",  "password": "12AASFTGJKLL",  "tutor": {    "noDocente": "AS34458816",    "horasTutoria": 12,    "carrera": 1  },  "fechaRegistro": "2025-05-13T19:14:41.911Z"}</pre>

Salidas:

Request URL	Request URL								
<code>http://127.0.0.1:8000/usuarios/privado/tutor</code>	<code>http://127.0.0.1:8000/usuarios/privado/tutor</code>								
Server response	Server response								
<table><tr><th>Code</th><th>Details</th></tr><tr><td>400</td><td>Error: Bad Request</td></tr></table> <div>Response body</div> <pre>{  "detail": "La contraseña debe contener al menos un número"}</pre>	Code	Details	400	Error: Bad Request	<table><tr><th>Code</th><th>Details</th></tr><tr><td>400</td><td>Error: Bad Request</td></tr></table> <div>Response body</div> <pre>{  "detail": "La contraseña debe contener al menos un carácter especial"}</pre>	Code	Details	400	Error: Bad Request
Code	Details								
400	Error: Bad Request								
Code	Details								
400	Error: Bad Request								

3. Terceras validaciones: Error, no existe la carrera especificada (ejemplo con coordinador)

POST /usuarios/privado/coordinador Registro para coordinadores (requiere autenticación)	Request URL				
Parameters	<code>http://127.0.0.1:8000/usuarios/privado/coordinador</code>				
No parameters	Server response				
Request body <small>required</small>	<table><tr><th>Code</th><th>Details</th></tr><tr><td>400</td><td>Error: Bad Request</td></tr></table> <div>Response body</div> <pre>{  "detail": "La carrera especificada no existe"}</pre>	Code	Details	400	Error: Bad Request
Code	Details				
400	Error: Bad Request				
<pre>{  "email": "coordinadoruser@example.com",  "nombre": "Beto",  "apellidos": "Díaz",  "tipo": "coordinador",  "password": "CD*1245789",  "coordinador": {    "noEmpleado": "CD22232425",    "departamento": "CIENCIAS DE LA COMPUTACIÓN",    "carrera": 0  },  "fechaRegistro": "2025-05-13T19:31:26.020Z"}</pre>					

Salida:

4. Cuartas Validaciones: Casos de éxito

Alumno, Entradas/Salidas:

POST /usuarios/publico/alumno Registro público para alumnos	
Parameters	
No parameters	Request URL
	http://127.0.0.1:8000/usuarios/publico/alumno
Request body required	Server response
{ "email": "alumnouser@example.com", "nombre": "Leonardo", "apellidos": "Barriga G", "tipo": "alumno", "password": "ls*M1548722", "alumno": { "noControl": "20031042", "semestre": 8, "carrera": 1 }, "fechaRegistro": "2025-05-13T19:41:36.407Z" }	Code
	Details
	201
	Response body
	{ "estatus": "OK", "mensaje": "El usuario con ID 6823a125f9304440ce512618 ha sido registrado exitosamente", "id_usuario": "6823a125f9304440ce512618" }

Tutor, Entradas/Salidas:

POST /usuarios/privado/tutor Registro para tutores (requiere autenticación)	
Parameters	
No parameters	Request URL
	http://127.0.0.1:8000/usuarios/privado/tutor
Request body required	Server response
{ "email": "tutoruser@example.com", "nombre": "Aron", "apellidos": "Rocha", "tipo": "tutor", "password": "s2AASFTGKLL*", "tutor": { "noDocente": "s34458816", "horasTutoria": 12, "carrera": 1 }, "fechaRegistro": "2025-05-13T19:14:41.911Z" }	Code
	Details
	201
	Response body
	{ "estatus": "OK", "mensaje": "El usuario con ID 6823a1a6f9304440ce512619 ha sido registrado exitosamente", "id_usuario": "6823a1a6f9304440ce512619" }

Coordinador. Entradas/Salidas:

POST /usuarios/privado/coordinador Registro para coordinadores (requiere autenticación)	
Parameters	
No parameters	Request URL
	http://127.0.0.1:8000/usuarios/privado/coordinador
Request body required	Server response
{ "email": "coordinadoruser@example.com", "nombre": "Beto", "apellidos": "Díaz", "tipo": "coordinador", "password": "CD*1245789", "coordinador": { "noEmpleado": "CD22232425", "departamento": "CIENCIAS DE LA COMPUTACIÓN", "carrera": 1 }, "fechaRegistro": "2025-05-13T19:31:26.020Z" }	Code
	Details
	201
	Response body
	{ "estatus": "OK", "mensaje": "El usuario con ID 6823a1e2f9304440ce51261a ha sido registrado exitosamente", "id_usuario": "6823a1e2f9304440ce51261a" }

Validaciones Particulares

Comenzando con Alumnos:

## 1. Correo registrado, Entrada/Salida:

<b>POST</b> /usuarios/publico/alumno Registro público para alumnos	<b>Request URL</b>								
<b>Parameters</b>	<code>http://127.0.0.1:8000/usuarios/publico/alumno</code>								
No parameters	<b>Server response</b>								
<b>Request body</b> <small>required</small>	<table><tr><th>Code</th><th>Details</th></tr><tr><td>400</td><td>Error: Bad Request</td></tr><tr><td></td><td><b>Response body</b></td></tr><tr><td></td><td><pre>{   "detail": "El correo electrónico ya está registrado" }</pre></td></tr></table>	Code	Details	400	Error: Bad Request		<b>Response body</b>		<pre>{   "detail": "El correo electrónico ya está registrado" }</pre>
Code	Details								
400	Error: Bad Request								
	<b>Response body</b>								
	<pre>{   "detail": "El correo electrónico ya está registrado" }</pre>								
<pre>{   "email": "alumnouser@example.com",   "nombre": "Leonardo",   "apellidos": "Barriga 6",   "tipo": "alumno",   "password": "1s*M1548722",   "alumno": {     "noControl": "20031042",     "semestre": 0,     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:41:36.407Z" }</pre>									

## 2. Tamaño de dígitos inválido del No. de Control, Entrada/Salida:

<b>POST</b> /usuarios/publico/alumno Registro público para alumnos	<b>Request URL</b>								
<b>Parameters</b>	<code>http://127.0.0.1:8000/usuarios/publico/alumno</code>								
No parameters	<b>Server response</b>								
<b>Request body</b> <small>required</small>	<table><tr><th>Code</th><th>Details</th></tr><tr><td>400</td><td>Error: Bad Request</td></tr><tr><td></td><td><b>Response body</b></td></tr><tr><td></td><td><pre>{   "detail": "El número de control debe tener exactamente 8 dígitos" }</pre></td></tr></table>	Code	Details	400	Error: Bad Request		<b>Response body</b>		<pre>{   "detail": "El número de control debe tener exactamente 8 dígitos" }</pre>
Code	Details								
400	Error: Bad Request								
	<b>Response body</b>								
	<pre>{   "detail": "El número de control debe tener exactamente 8 dígitos" }</pre>								
<pre>{   "email": "betouser@example.com",   "nombre": "Carlos Humberto",   "apellidos": "Garcia",   "tipo": "alumno",   "password": "1s*M1548722",   "alumno": {     "noControl": "2",     "semestre": 8,     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:41:36.407Z" }</pre>									

## 3. Semestre inválido, Entrada/Salida:

<b>POST</b> /usuarios/publico/alumno Registro público para alumnos	<b>Server response</b>								
<b>Parameters</b>	<table><tr><th>Code</th><th>Details</th></tr><tr><td>400</td><td>Error: Bad Request</td></tr><tr><td></td><td><b>Response body</b></td></tr><tr><td></td><td><pre>{   "detail": "El semestre debe estar entre 1 y 12" }</pre></td></tr></table>	Code	Details	400	Error: Bad Request		<b>Response body</b>		<pre>{   "detail": "El semestre debe estar entre 1 y 12" }</pre>
Code	Details								
400	Error: Bad Request								
	<b>Response body</b>								
	<pre>{   "detail": "El semestre debe estar entre 1 y 12" }</pre>								
<b>Request body</b> <small>required</small>									
<pre>{   "email": "betouser@example.com",   "nombre": "Carlos Humberto",   "apellidos": "Garcia",   "tipo": "alumno",   "password": "1s*M1548722",   "alumno": {     "noControl": "20031034",     "semestre": 0,     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:41:36.407Z" }</pre>									

## 4. Carrera Inexistente:

<b>POST</b> /usuarios/publico/alumno Registro público para alumnos	<b>Request URL</b> http://127.0.0.1:8000/usuarios/publico/alumno				
<b>Parameters</b> No parameters	<b>Server response</b>				
<b>Request body</b> <small>required</small> <pre>{   "email": "betouser@example.com",   "nombre": "Carlos Humberto",   "apellidos": "García",   "tipo": "alumno",   "password": "1s*M1548722",   "alumno": {     "noControl": "20232125",     "semestre": 8,     "carrera": 0   },   "fechaRegistro": "2025-05-13T19:41:36.407Z" }</pre>	<table> <tr> <th>Code</th><th>Details</th></tr> <tr> <td>400</td><td> <b>Error: Bad Request</b>  <b>Response body</b>  <pre>{   "detail": "La carrera especificada no existe" }</pre> </td></tr> </table>	Code	Details	400	<b>Error: Bad Request</b> <b>Response body</b> <pre>{   "detail": "La carrera especificada no existe" }</pre>
Code	Details				
400	<b>Error: Bad Request</b> <b>Response body</b> <pre>{   "detail": "La carrera especificada no existe" }</pre>				

## 5. Número de control existente, Entrada/Salida:

<b>POST</b> /usuarios/publico/alumno Registro público para alumnos	<b>Request URL</b> http://127.0.0.1:8000/usuarios/publico/alumno				
<b>Parameters</b> No parameters	<b>Server response</b>				
<b>Request body</b> <small>required</small> <pre>{   "email": "betouser@example.com",   "nombre": "Carlos Humberto",   "apellidos": "García",   "tipo": "alumno",   "password": "1s*M1548722",   "alumno": {     "noControl": "20031042",     "semestre": 8,     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:41:36.407Z" }</pre>	<table> <tr> <th>Code</th><th>Details</th></tr> <tr> <td>400</td><td> <b>Error: Bad Request</b>  <b>Response body</b>  <pre>{   "detail": "El número de control ya está registrado" }</pre> </td></tr> </table>	Code	Details	400	<b>Error: Bad Request</b> <b>Response body</b> <pre>{   "detail": "El número de control ya está registrado" }</pre>
Code	Details				
400	<b>Error: Bad Request</b> <b>Response body</b> <pre>{   "detail": "El número de control ya está registrado" }</pre>				

## Validaciones específicas para el Tutor:

1. El correo del docente ya está registrado:

POST	/usuarios/privado/tutor	Registro para tutores (requiere autenticación)
Parameters		
No parameters		
Request body <small>required</small>		
<pre>{   "email": "tutouser@example.com",   "nombre": "Aron",   "apellidos": "Rocha",   "tipo": "tutor",   "password": "12AASFTGJKLL*",   "tutor": {     "noDocente": "5344",     "horasTutoria": 12,     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:14:41.911Z" }</pre>		

#### Server response

##### Code Details

400 Error: Bad Request

##### Response body

```
{
  "detail": "El correo electrónico ya está registrado"
}
```

## 2. Restricción en la entrada de dígitos para el número de docente, Entrada/Salida:

POST	/usuarios/privado/tutor	Registro para tutores (requiere autenticación)
Parameters		
No parameters		
Request body <small>required</small>		
<pre>{   "email": "tutouser@example.com",   "nombre": "Aron",   "apellidos": "Rocha",   "tipo": "tutor",   "password": "12AASFTGJKLL*",   "tutor": {     "noDocente": "5344",     "horasTutoria": 12,     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:14:41.911Z" }</pre>		

#### Request URL

http://127.0.0.1:8000/usuarios/privado/tutor

#### Server response

##### Code Details

400 Error: Bad Request

##### Response body

```
{
  "detail": "El número de docente debe comenzar con letra seguida de 4 a 9 dígitos"
}
```

## 3. Las horas de tutoría deben estar entre 1 y 40:

POST	/usuarios/privado/tutor	Registro para tutores (requiere autenticación)
Parameters		
No parameters		
Request body <small>required</small>		
<pre>{   "email": "tutouser@example.com",   "nombre": "Aron",   "apellidos": "Rocha",   "tipo": "tutor",   "password": "12AASFTGJKLL*",   "tutor": {     "noDocente": "5344556639",     "horasTutoria": 0,     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:14:41.911Z" }</pre>		

#### Request URL

http://127.0.0.1:8000/usuarios/privado/tutor

#### Server response

##### Code Details

400 Error: Bad Request

##### Response body

```
{
  "detail": "Las horas de tutoría deben estar entre 1 y 40"
}
```

## 4. La carrera especificada no existe:

#### Request URL

http://127.0.0.1:8000/usuarios/privado/tutor

#### Server response

##### Code Details

400 Error: Bad Request

##### Response body

```
{
  "detail": "La carrera especificada no existe"
}
```

POST	/usuarios/privado/tutor	Registro para tutores (requiere autenticación)
Parameters		
No parameters		
Request body <small>required</small>		
<pre>{   "email": "tutouser@example.com",   "nombre": "Aron",   "apellidos": "Rocha",   "tipo": "tutor",   "password": "12AASFTGJKLL*",   "tutor": {     "noDocente": "5344556639",     "horasTutoria": 23,     "carrera": 0   },   "fechaRegistro": "2025-05-13T19:14:41.911Z" }</pre>		

5. El número de docente ya está registrado:

POST	/usuarios/privado/tutor	Registro para tutores (requiere autenticación)
Parameters		
No parameters		
Request body <small>required</small>		
<pre>{   "email": "tutouser@example.com",   "nombre": "Aron",   "apellidos": "Rocha",   "tipo": "tutor",   "password": "12AASFTGJKLL*",   "tutor": {     "noDocente": "5344558816",     "horasTutoria": 23,     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:14:41.911Z" }</pre>		
POST	/usuarios/privado/coordinador	Registro para coordinadores (requiere autenticación)
Parameters		
No parameters		
Request body <small>required</small>		
<pre>{   "email": "coordinadoruser@example.com",   "nombre": "Beto",   "apellidos": "Díaz",   "tipo": "coordinador",   "password": "CD1245789",   "coordinador": {     "noEmpleado": "CD22232425",     "departamento": "CIENCIAS DE LA COMPUTACIÓN",     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:31:26.020Z" }</pre>		

ordinador:  
está registra

Request URL	
http://127.0.0.1:8000/usuarios/privado/tutor	
Server response	
Code	Details
400	Error: Bad Request
Response body	
{   "detail": "El número de docente ya está registrado" }	
Request URL	
http://127.0.0.1:8000/usuarios/privado/coordinador	
Server response	
Code	Details
400	Error: Bad Request
Response body	
{   "detail": "El correo electrónico ya está registrado" }	

2. Restricción en la entrada de dígitos para el número del empleado:

POST	/usuarios/privado/coordinador	Registro para coordinadores (requiere autenticación)
Parameters		
No parameters		
Request body <small>required</small>		
<pre>{   "email": "coordinadoruser@example.com",   "nombre": "Beto",   "apellidos": "Díaz",   "tipo": "coordinador",   "password": "CD1245789",   "coordinador": {     "noEmpleado": "C225",     "departamento": "CIENCIAS DE LA COMPUTACIÓN",     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:31:26.020Z" }</pre>		

Request URL	
http://127.0.0.1:8000/usuarios/privado/coordinador	
Server response	
Code	Details
400	Error: Bad Request
Response body	
{   "detail": "El número de empleado debe comenzar con 2 letras seguidas de 3 a 8 dígitos" }	

3. El departamento debe tener entre 3 y 50 caracteres:



POST	/usuarios/privado/coordinador	Registro para coordinadores (requiere autenticación)
Parameters		
No parameters		
Request body <i>required</i>		
<pre>{   "email": "corduser@example.com",   "nombre": "Beto",   "apellidos": "Díaz",   "tipo": "coordinador",   "password": "CD*1245789",   "coordinador": {     "noEmpleado": "CC225",     "departamento": "CC",     "carrera": 1   },   "fechaRegistro": "2025-05-13T19:31:26.020Z" }</pre>		

Request URL	
http://127.0.0.1:8000/usuarios/privado/coordinador	
Server response	
Code	Details
400	Error: Bad Request
Response body	
<pre>{   "detail": "El departamento debe tener entre 3 y 50 caracteres" }</pre>	

#### 4. La carrera especificada no existe:

POST	/usuarios/privado/coordinador	Registro para coordinadores (requiere autenticación)
Parameters		
No parameters		
Request body <span>required</span>		
<pre>{  "email": "corduser@example.com",  "nombre": "Beto",  "apellidos": "Díaz",  "tipo": "coordinador",  "password": "CD*1245789",  "coordinador": {    "noEmpleado": "CC225",    "departamento": "CIENCIAS DE LA COMP",    "carrera": 0  },  "fechaRegistro": "2025-05-13T19:31:26.020Z"}</pre>		

Request URL	
http://127.0.0.1:8000/usuarios/privado/coordinador	
Server response	
Code	Details
400	Error: Bad Request
Response body	
<pre>{  "detail": "La carrera especificada no existe"}</pre>	

#### 5. El número de empleado ya está registrado:

POST /usuarios/privado/coordinador Registro para coordinadores (requiere autenticación)	
Parameters	
No parameters	
Request body <span>required</span>	
<pre>{  "email": "corduser@example.com",  "nombre": "Beto",  "apellidos": "Díaz",  "tipo": "coordinador",  "password": "CD*1245789",  "coordinador": {    "noEmpleado": "CD2232425",    "departamento": "CIENCIAS DE LA COMP",    "carrera": 1  },  "fechaRegistro": "2025-05-13T19:31:26.020Z"}</pre>	
Request URL	
http://127.0.0.1:8000/usuarios/privado/coordinador	
Server response	
Code	Details
400	Error: Bad Request
Response body	
<pre>{  "detail": "El número de empleado ya está registrado"}</pre>	

## Consultar general de usuarios

GET /usuarios/general/ Consulta general de usuarios

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/usuarios/general/' \
-H 'accept: application/json'
```

Respuesta correcta:

Request URL

http://127.0.0.1:8000/usuarios/general/

Server response

Code	Details
200	<p>Response body</p> <pre>{   "estatus": "OK",   "mensaje": "Se encontraron 4 usuarios",   "usuarios": [     {       "id": "6823a125f9304440ce512618",       "email": "alumno@example.com",       "nombre": "Juan",       "apellidos": "Pérez",       "tipo": "alumno",       "fechaRegistro": "2025-05-13T19:41:36.407000",       "alumno": {         "noControl": "20230001",         "semestre": 5,         "carrera": 1       },       "nombreCarrera": "Sin carrera asignada"     },     {       "id": "6823a1a6f9304440ce512619",       "email": "tutor@example.com",       "nombre": "Maria",       "apellidos": "Gómez",       "tipo": "tutor",       "fechaRegistro": "2025-05-13T19:14:41.911000",       "tutor": {         "noDocente": "T12345",         "horasTutoria": 10,         "carrera": 1       }     }   ] }</pre>

Error:

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/usuarios/general/' \
-H 'accept: application/json'
```

Request URL

http://127.0.0.1:8000/usuarios/general/

Server response

Code	Details
200	<p>Response body</p> <pre>{   "estatus": "ERROR",   "mensaje": "Error al obtener el listado de usuarios",   "usuarios": [] }</pre>

## Consultar de actividad por Id

GET

/usuarios/{id\_usuario} Consultar un usuario por su ID

^

Parameters

Cancel

Name	Description
id_usuario <span>required</span>	
string (path)	6823a125f9304440ce512618

Execute

Respuesta Correcta:

Request URL

http://127.0.0.1:8000/usuarios/6823a125f9304440ce512618

Server response

Code	Details
200	<div>Response body</div> <pre>{   "estatus": "OK",   "mensaje": "Usuario encontrado exitosamente",   "usuario": {     "id": "6823a125f9304440ce512618",     "email": "alumno@example.com",     "nombre": "Juan",     "apellidos": "Pérez",     "tipo": "alumno",     "fechaRegistro": "2025-05-13T19:41:36.407000",     "alumno": {       "noControl": "20230001",       "semestre": 5,       "carrera": 1     },     "nombreCarrera": null   } }</pre>

Error:

Request URL

http://127.0.0.1:8000/usuarios/6823a125f9304440ce512618

Server response

Code	Details
500	<div>Error: Internal Server Error</div> <div>Response body</div> <pre>{   "detail": "Error interno al consultar el usuario" }</pre>

## Actualizar usuario (ejemplo con coordinador)

PUT	/usuarios/coordinador/{id_usuario}	Actualizar información de un coordinador
Parameters		
Name	Description	
id_usuario * required		
string		6823a1e2f9304440ce51261a
(path)		
Request body required		
<pre>{   "nombre": "Carlos",   "apellidos": "López",   "email": "coordinador@example.com",   "password": "NuevaContraseña123",   "coordinador": {     "carrera": 1,     "departamento": "Tutorías",     "noEmpleado": "C012345",     "nombreCarrera": "Ingeniería en Sistemas"   } }</pre>		

Respuesta Correcta:

Request URL	
http://127.0.0.1:8000/usuarios/coordinador/6823a1e2f9304440ce51261a	
Server response	
Code	Details
200	Response body
	<pre>{   "estatus": "OK",   "mensaje": "Coordinador actualizado correctamente." }</pre>

Error: Usando el id de un Tutor

Request URL	
http://127.0.0.1:8000/usuarios/coordinador/682449a3dd93f72cac59e908	
Server response	
Code	Details
404	Error: Not Found
	Response body
	<pre>{   "detail": "El usuario no existe o no es un coordinador" }</pre>

## Eliminar usuario

Se pide el id del usuario a eliminar y el número de empleado valido (el coordinador es el único rol que puede eliminar usuarios)

**DELETE** /usuarios/{id\_usuario} Eliminar un usuario permanentemente

**Parameters**

Name	Description
id_usuario * required string (path)	<input type="text" value="682449a3dd93f72cac59e908"/>

**Request body** required

```
{  
  "no_empleado_coordinador": "C012345"  
}
```

Respuesta correcta:

Code	Description
200	Usuario eliminado con éxito

Media type

Controls Accept header.

Example Value | Schema

```
{  
  "mensaje": "string",  
  "detalles Eliminacion": {  
    "additionalProp1": {}  
  },  
  "coordinador_autorizador": "string",  
  "fecha_eliminacion": "2025-05-14T13:37:27.662Z"  
}
```

Error:

Request URL  
`http://127.0.0.1:8000/usuarios/682449a3dd93f72cac59e908`

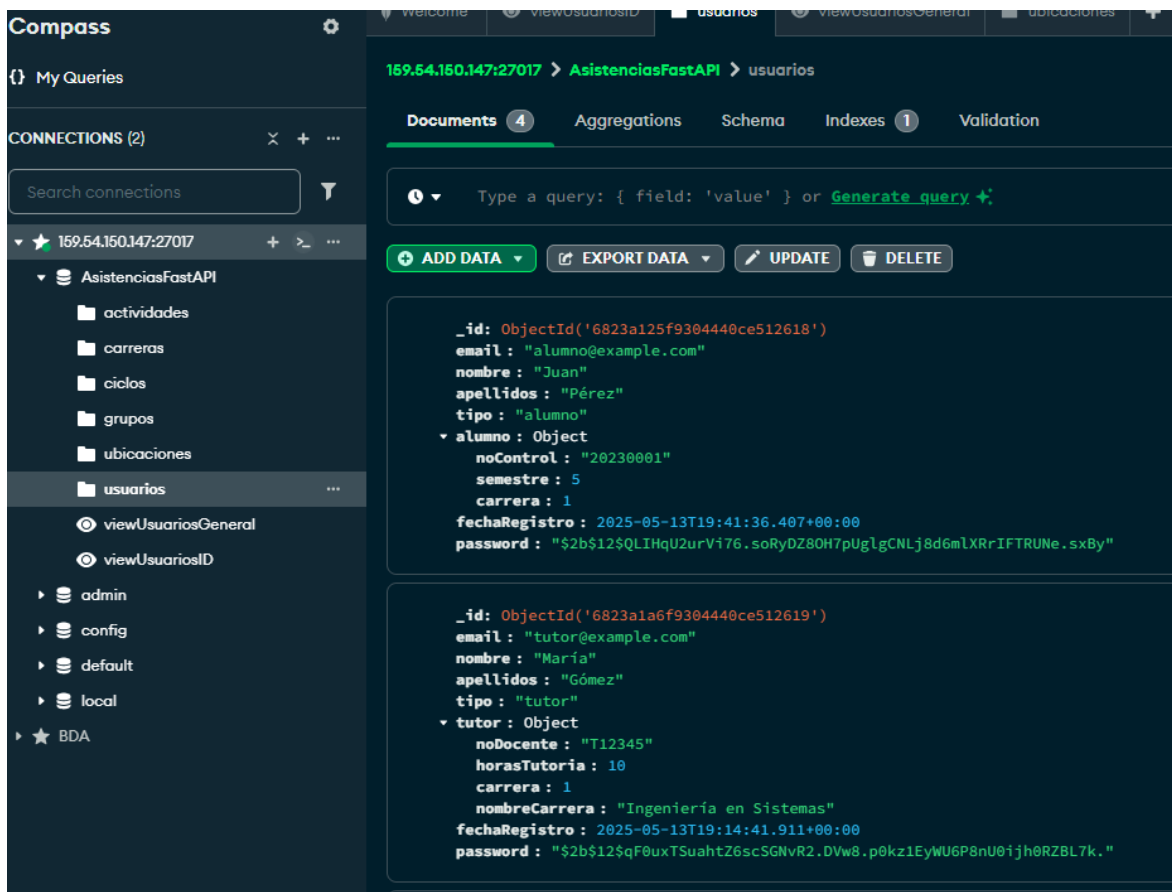
Server response

Code	Details
403	Error: Forbidden

Response body  

```
{  
  "detail": "Acceso denegado: Solo coordinadores pueden eliminar usuarios"  
}
```

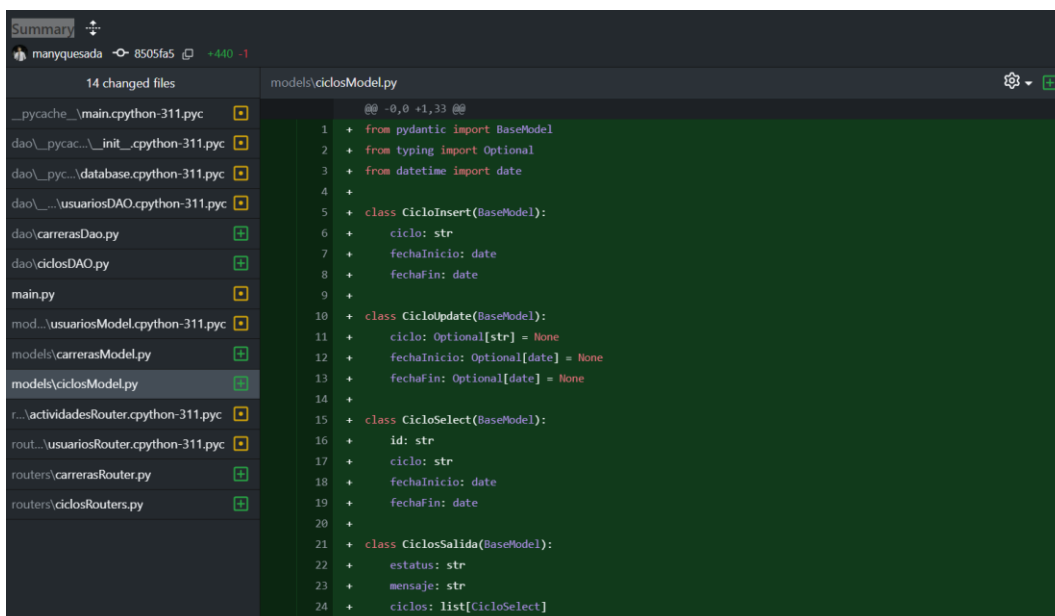
## Comprobación MongoDB



## Ciclos

### Commits

Se realizaron los siguientes commits en realizados Carlos Emmanuel López Quesada Git referente al servicio de ciclos tanto carreras , que se puede resumir en las siguientes rutas, operaciones, salidas.



The screenshot shows a Git commit summary for user 'manyquesada' with commit hash '8505fa5'. It lists 14 changed files, including various Python files in the 'dao' and 'models' directories. The main part of the image shows a code diff for 'models/ciclosModel.py'. The code defines three classes: 'CicloInsert', 'CicloUpdate', and 'CicloSelect', all inheriting from 'BaseModel'. 'CicloInsert' has attributes 'ciclo' (str), 'fechaInicio' (date), and 'fechaFin' (date). 'CicloUpdate' has 'ciclo' (Optional[str]), 'fechaInicio' (Optional[date]), and 'fechaFin' (Optional[date]). 'CicloSelect' has 'id' (str), 'ciclo' (str), 'fechaInicio' (date), and 'fechaFin' (date). There is also a 'CiclosSalida' class with 'estatus' (str), 'mensaje' (str), and 'ciclos' (list[CicloSelect]).

Rutas del servicio en swagger:

#### Ciclos

GET	/ciclos/	Consultar todos los ciclos
POST	/ciclos/	Crear un nuevo ciclo
GET	/ciclos/{idCiclo}	Consultar un ciclo por su ID
PUT	/ciclos/{idCiclo}	Actualizar un ciclo
DELETE	/ciclos/{idCiclo}	Eliminar un ciclo

### Registrar ciclos




The Swagger UI for the 'Registrar ciclos' endpoint shows the following details:

- Parameters:** No parameters.
- Request body:** Required, with a dropdown menu set to 'application/json'.
- Example Value:** A JSON object: 

```
{  "ciclo": "string",  "fechaInicio": "2025-05-14",  "fechaFin": "2025-05-14"}
```


## Ejemplo de error:

Code	Details
201	<div><div>Response body</div><div><pre>{   "estatus": "ERROR",   "mensaje": "Uno o más alumnos especificados no existen o no son de tipo alumno",   "grupo": null }</pre></div><div> Download</div></div>

## Consulta general de ciclos

Parameters			Try it out
No parameters			
Responses			
Code	Description	Links	
200	Successful Response	No links	
<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value   Schema</div>			
<pre>{   "estatus": "string",   "mensaje": "string",   "ciclos": [     {       "id": "string",       "ciclo": "string",       "fechaInicio": "2025-05-14",       "fechaFin": "2025-05-14"     }   ] }</pre>			

## Respuesta de agregado correctamente:

Code	Details
200	<div><div>Response body</div><div><pre>{   "estatus": "OK",   "mensaje": "Lista de ciclos obtenida correctamente.",   "ciclos": [     {       "id": "68245cde5ac87884f230de25",       "ciclo": "SEP",       "fechaInicio": "2025-05-14",       "fechaFin": "2025-05-14"     },     {       "id": "68245cff5ac87884f230de26",       "ciclo": "SEP-DEC",       "fechaInicio": "2026-05-14",       "fechaFin": "2026-12-14"     },     {       "id": "68245d015ac87884f230de27",       "ciclo": "SEP-DEC",       "fechaInicio": "2026-05-14",       "fechaFin": "2026-12-14"     },     {       "id": "68245d025ac87884f230de28",       "ciclo": "SEP-DEC",       "fechaInicio": "2026-05-14",       "fechaFin": "2026-12-14"     }   ] }</pre></div><div> Download</div></div>



## Consulta un ciclo por su id

**GET** `/ciclos/{idCiclo}` Consultar un ciclo por su ID

Obtiene un ciclo académico específico por su ID.

**Parameters** Try it out

Name	Description
<b>idCiclo</b> <small>required</small>	
string (path)	<input type="text" value="idCiclo"/>

**Responses**

Code	Description	Links
200	Successful Response	No links

Media type:

Controls Accept header.

Example Value | Schema

```
{
  "estatus": "string",
  "mensaje": "string",
  "ciclo": {
    "id": "string",
    "ciclo": "string",
    "fechaInicio": "2025-05-14",
    "fechaFin": "2025-05-14"
  }
}
```

Respuesta de agregado correctamente:

**idCiclo** required

string  
(path)

Execute Clear

**Responses**

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/ciclos/68245cff5ac87884f230de26' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/ciclos/68245cff5ac87884f230de26
```

Server response

Code	Details
200	<div>Response body</div> <pre>{   "estatus": "OK",   "mensaje": "Ciclo obtenido correctamente.",   "ciclo": {     "id": "68245cff5ac87884f230de26",     "ciclo": "SEP-DEC",     "fechaInicio": "2026-05-14",     "fechaFin": "2026-12-14"   } }</pre> <div><span>Download</span></div>

Ejemplo de error:

Code	Details
200	<div>Response body</div> <pre>{   "estatus": "ERROR",   "mensaje": "Ciclo no encontrado.",   "ciclo": null }</pre> <div><span>Download</span></div> <div>Response headers</div> <pre>content-length: 65 content-type: application/json date: Wed, 14 May 2025 14:53:58 GMT server: uvicorn</pre>

## Actualizar Ciclo

**PUT** `/ciclos/{idCiclo}` Actualizar un ciclo

Actualiza la información de un ciclo académico existente.

**Parameters** Try it out

Name	Description
<b>idCiclo</b> <span>required</span>	
string	idCiclo
(path)	

**Request body** required application/json

**Example Value** **Schema**

```
{
  "ciclo": "string",
  "fechaInicio": "2025-05-14",
  "fechaFin": "2025-05-14"
}
```

Respuesta de agregado correctamente:

201

**Response body**

```
{
  "estatus": "OK",
  "mensaje": "Grupo creado con ID: 682433dbe6e6e5c5d49915a4",
  "grupo": {
    "id": "682433dbe6e6e6e5c5d49915a4",
    "nombre": "A",
    "semestre": 8,
    "ciclo": "67d5069a5c4a73aca45548b8",
    "carrera": 1,
    "tutor": "6823a1a6f9304440ce512619",
    "alumnos": [
      "6823a125f9304440ce512618"
    ]
  }
}
```

 **Download**


Ejemplo de error:

**Code** **Details**

200

**Response body**

```
{
  "estatus": "ERROR",
  "mensaje": "Error al actualizar el ciclo: Invalid document {'q': {'_id': ObjectId('68245cff5ac87884f230de26')}, 'u': {'$set': {'ciclo': 'Sep', 'fechaInicio': datetime.date(2025, 5, 14), 'fechaFin': datetime.date(2025, 5, 14)}}, 'multi': False, 'upsert': False} ] cannot encode object: datetime.date(2025, 5, 14), of type: <class 'datetime.date'>",
  "ciclo": null
}
```

 **Download**

**Response headers**

```
content-length: 378
content-type: application/json
date: Wed, 14 May 2025 14:57:12 GMT
server: uvicorn
```

## Eliminar Ciclo

**DELETE** /ciclos/{idCiclo} Eliminar un ciclo

Elimina un ciclo académico.

**Parameters** Try it out

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

**Responses**

Code	Description	Links
200	Successful Response <div>Media type <div>application/json</div><div>Controls Accept header.</div><div>Example Value   Schema</div><pre>{   "estatus": "string",   "mensaje": "string" }</pre></div>	No links

Respuesta de agregado correctamente:

**DELETE** /ciclos/{idCiclo} Eliminar un ciclo

Elimina un ciclo académico.

**Parameters** Cancel

Name	Description
<b>idCiclo</b> * required string (path)	<input type="text" value="68245cff5ac87884f230de25"/>

Execute

Clear

**Responses**

**Curl**

```
curl -X DELETE \
  http://127.0.0.1:3000/ciclos/{idCiclo} \
  -H 'accept: application/json'
```

**Request URL**

```
http://127.0.0.1:3000/ciclos/68245cff5ac87884f230de25
```

**Server response**

Code	Value
200	<div><b>Response body</b><pre>{   "estatus": "OK",   "mensaje": "Ciclo eliminado correctamente." }</pre></div> <div><b>Response headers</b><pre>Content-Length: 62 Content-Type: application/json Date: Wed, 14 May 2025 14:59:00 GMT Server: uvicorn</pre></div>

Ejemplo de error:

Code	Details
200	<div><b>Response body</b><pre>{   "estatus": "ERROR",   "mensaje": "Ciclo con ID 68245cff5ac87884f230de25 no encontrado." }</pre></div> <div><b>Response headers</b><pre>content-length: 84 content-type: application/json date: Wed, 14 May 2025 14:59:00 GMT server: uvicorn</pre></div>

# Comprobación MongoDB

Antes de actualizar

<a href="#">ADD DATA</a>	<a href="#">EXPORT DATA</a>	<a href="#">UPDATE</a>	<a href="#">DELETE</a>	25	1 - 11 of 11	<a href="#">↺</a>	<a href="#">↻</a>	<a href="#">↷</a>	<a href="#">☰</a>	<a href="#">🔍</a>
<pre>{   "_id": ObjectId("68245cff5ac87884f238de26"),   "ciclo": "SEP-DEC",   "fechaInicio": "2026-05-14",   "fechaFin": "2026-12-14" }</pre>										
<pre>{   "_id": ObjectId("68245d015ac87884f238de27"),   "ciclo": "SEP-DEC",   "fechaInicio": "2026-05-14",   "fechaFin": "2026-12-14" }</pre>										
<pre>{   "_id": ObjectId("68245d025ac87884f238de28"),   "ciclo": "SEP-DEC",   "fechaInicio": "2026-05-14",   "fechaFin": "2026-12-14" }</pre>										

Despues de actualizar

<a href="#">ADD DATA</a>	<a href="#">EXPORT DATA</a>	<a href="#">UPDATE</a>	<a href="#">DELETE</a>	25	1 - 10 of 10	<a href="#">↺</a>	<a href="#">↻</a>	<a href="#">↷</a>	<a href="#">☰</a>	<a href="#">🔍</a>
<pre>{   "_id": ObjectId("68245cde5ac87884f238de25"),   "ciclo": "SEP",   "fechaInicio": "2025-05-14",   "fechaFin": "2025-05-14" }</pre>										
<pre>{   "_id": ObjectId("68245d015ac87884f238de27"),   "ciclo": "SEP-DEC",   "fechaInicio": "2026-05-14",   "fechaFin": "2026-12-14" }</pre>										
<pre>{   "_id": ObjectId("68245d025ac87884f238de28"),   "ciclo": "SEP-DEC",   "fechaInicio": "2026-05-14",   "fechaFin": "2026-12-14" }</pre>										

## Carreras

Rutas del servicio en swagger:

Carreras			^
GET	/carreras/	Consultar todas las carreras	▼
POST	/carreras/	Crear una nueva carrera	▼
GET	/carreras/{carrera_id}	Consultar una carrera por su ID	▼
PUT	/carreras/{carrera_id}	Actualizar una carrera	▼
DELETE	/carreras/{carrera_id}	Eliminar una carrera	▼

## Registrar carrera

**POST** /carreras/ Crear una nueva carrera ^

Parameters Try it out

No parameters

Request body <sup>required</sup> application/json ▼


Example Value | Schema

```
{
  "carrera": "string",
  "nombre": "string"
}
```

Respuesta de agregado correctamente:

```
{
  "_id": ObjectId('68248ae666f66b35e9171703'),
  "carrera": "CB",
  "nombre": "CiberSeguridad"
}
```

Ejemplo de error:

Code	Details
201	<div>Response body</div> <div><pre>{   "estatus": "ERROR",   "mensaje": "Error interno al agregar la carrera.",   "carrera": null }</pre></div> <div> Download</div>

## Consulta general de carreras

GET

/carreras/ Consultar todas las carreras

⌵

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/carreras/' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/carreras/
```

Server response

Respuesta de agregado correctamente:

Ejemplo de error:

Code	Details
201	<div>Response body<div><pre>{   "estatus": "ERROR",   "mensaje": "Uno o más alumnos especificados no existen o no son de tipo alumno",   "grupo": null }</pre></div><div><div></div>Download</div></div>

## Consulta un carrera por su id

GET

/carreras/{carrera\_id} Consultar una carrera por su ID

⌵

Parameters

Cancel

Name

Description

carrera\_id required

integer

(path)

1

Execute

Responses

Code	Description	Links
200	<div>Successful Response</div> <div>Media type<div>application/json</div></div> <div>Controls Accept header.</div> <div>Example Value   Schema<div><pre>{   "estatus": "string",   "mensaje": "string",   "carrera": {     "id": 0,     "carrera": "string",     "nombre": "string"   } }</pre></div></div>	

 No links |

Respuesta de agregado correctamente:

Server response	
Code	Details
200	<div>Response body</div> <pre>{   "estatus": "OK",   "mensaje": "Carrera encontrada.",   "carrera": {     "id": 3,     "carrera": "CP",     "nombre": "Contaduría Pública"   } }</pre> <div>Download</div> <div>Response headers</div> <pre>content-length: 114 content-type: application/json date: Wed, 14 May 2025 15:04:16 GMT server: uvicorn</pre>

Ejemplo de error:

Code	Details
200	<div>Response body</div> <pre>{   "estatus": "ERROR",   "mensaje": "Carrera con ID 7 no encontrada.",   "carrera": null }</pre> <div>Download</div>

## Actualizar Carrera

PUT /carreras/{carrera\_id} Actualizar una carrera

Try it out

Name	Description
carrera_id * required integer (path)	<input type="text" value="1"/>

Request body required application/json

Example Value | Schema

```
{
  "carrera": "string",
  "nombre": "string"
}
```

Respuesta de agregado correctamente:

Code	Details
200	<div>Response body</div> <pre>{   "estatus": "OK",   "mensaje": "Carrera actualizada correctamente.",   "carrera": {     "id": 3,     "carrera": "CS",     "nombre": "Contaduría Pública"   } }</pre> <div>Download</div>

Ejemplo de error:

Code	Details
200	<div>Response body</div> <pre>{   "estatus": "ERROR",   "mensaje": "Carrera con ID 7 no encontrada.",   "carrera": null }</pre> <div>Download</div>

## Eliminar Carrera

**DELETE** /carreras/{carrera\_id} Eliminar una carrera

Try it out

Name	Description
<b>carrera_id</b> * required integer (path)	<input type="text" value="carrera_id"/>

Responses

Code	Description	Links
200	Successful Response	No links

Media type

Controls Accept header.

Example Value | Schema

```
{  "estatus": "string",  "mensaje": "string",  "carrera": {    "id": 0,    "carrera": "string",    "nombre": "string"  }}
```

Respuesta de agregado correctamente:

**carrera\_id** \* required  
integer  
(path)

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \  'http://127.0.0.1:8000/carreras/3' \  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/carreras/3
```

Server response

Code	Details
200	Response body

```
{  "estatus": "OK",  "mensaje": "Carrera eliminada exitosamente. ID: 3",  "carrera": null}
```

Download

Ejemplo de error:

**carrera\_id** \* required  
integer  
(path)

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \  'http://127.0.0.1:8000/carreras/7' \  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/carreras/7
```

Server response

Code	Details
200	Response body

```
{  "estatus": "ERROR",  "mensaje": "Carrera con ID 7 no encontrada.",  "carrera": null}
```

Download

Response headers

```
content-length: 78  content-type: application/json  date: Wed, 18 Mar 2020 15:11:06 GMT  server: uvicorn
```



## Comprobación MongoDB

### Antes de actualizar

```
_id: 2  
carrera: "IIA"  
nombre: "Ingeniería Industrias Alimentarias"
```

```
_id: 3  
carrera: "CP"  
nombre: "Contaduría Pública"
```

```
_id: 4  
carrera: "II"  
nombre: "Ingeniería Industrial"
```

### Después de actualizar

```
_id: 3  
carrera: "C"  
nombre: "Contaduría Pública"
```