# ACTIVIDAD SOA

Universidad: Universidad Cristóbal Colón

Nombre del alumno: Marco Antonio Lagunes Montero

Carrera: Ingeniería en Sistemas Computacionales

Profesor: Leonardo.

Materia: SOA (Arquitectura Orientada a Servicios)

Proyecto: Actividad de clase 21/11/2025
Fecha: 21 de noviembre de 2025

## Introducción

En esta actividad se desarrolló una pequeña plataforma académica basada en Arquitectura Orientada a Servicios (SOA). El objetivo principal fue implementar dos tipos de servicios web que consumen la misma base de datos MySQL:
• Una API RESTful desarrollada con Java y Spring Boot.
• Un servicio SOAP desarrollado con Python y la librería Spyne.

Ambas soluciones trabajan sobre el mismo esquema de base de datos denominado soa_universidad, lo que permite demostrar cómo distintos estilos de servicios pueden convivir, reutilizar la misma información y ser consumidos desde clientes como Postman.

## Diseño de la base de datos soa_universidad

La base de datos utilizada para ambos servicios se llama soa_universidad y contiene las siguientes tablas principales:

• alumnos (id, nombre, email, carrera)
• calificaciones (id, alumno_id, materia, calificacion, fecha)
• matriculas (id, alumno_id, periodo, estatus)

Las tablas calificaciones y matriculas referencian a alumnos mediante la llave foránea alumno_id, garantizando la integridad entre los registros.

## Implementación de la API REST (Spring Boot)

La API REST se desarrolló con Java y Spring Boot, creando entidades JPA, repositorios y controladores REST. Los controladores principales fueron AlumnoController y CalificacionController, expuestos bajo el contexto /api.

### Endpoints de alumnos

Ruta base: /api/alumnos

• GET /api/alumnos – Lista todos los alumnos.
• POST /api/alumnos – Crea un nuevo alumno, recibiendo un JSON como:

```
{
  "nombre": "Marco Antonio Lagunes Montero",
  "email": "marcolagunes@ucc.mx",
  "carrera": "Ciberseguridad"
}
```

• PUT /api/alumnos/{id} – Actualiza datos de un alumno.
• DELETE /api/alumnos/{id} – Elimina un alumno por ID.

### Endpoints de calificaciones

Ruta base: /api/calificaciones

• GET /api/calificaciones – Lista todas las calificaciones.
• POST /api/calificaciones – Crea una calificación ligada a un alumno.
• PUT /api/calificaciones/{id} – Actualiza una calificación existente.
• DELETE /api/calificaciones/{id} – Elimina una calificación por ID.

Todas las pruebas se hicieron con Postman verificando códigos 200/201 y mensajes de error adecuados en caso de datos inexistentes.

## Implementación del servicio SOAP (Python + Spyne)

El servicio SOAP se implementó en Python usando Spyne y mysql-connector-python. El servicio MatriculaService expone operaciones para consultar y administrar las matrículas de la tabla matriculas. El servidor se ejecuta en:

http://localhost:8000/

y el WSDL está disponible en:

http://localhost:8000/?wsdl

### Operaciones de MatriculaService

• getMatricula(matricula_id: int) – Devuelve los datos de una matrícula por ID.
• createMatricula(alumno_id: int, periodo: string, estatus: string) – Inserta una nueva matrícula.
• getAllMatriculas() – Regresa todas las matrículas en cadenas con el formato id|alumno_id|periodo|estatus.

Las peticiones se enviaron desde Postman con método POST, cuerpo XML y encabezado Content-Type: text/xml.

## Evidencias de funcionamiento

Se realizaron pruebas en Postman para verificar el funcionamiento de las APIs:

• REST – GET /api/alumnos: respondió con la lista de alumnos en JSON (200 OK).
• REST – POST /api/alumnos: creó un nuevo alumno y se verificó en la base de datos.
• REST – POST /api/calificaciones: registró calificaciones ligadas a alumnos existentes.
• SOAP – getAllMatriculas: devolvió varias etiquetas <tns:string> con el contenido de las matrículas (por ejemplo 1|1|2025-1|Activo).
• SOAP – createMatricula: después de la respuesta de éxito, la fila apareció en la tabla

matriculas de MySQL Workbench.

En el documento final se pueden insertar capturas de pantalla en este apartado como evidencia visual (Postman, Workbench y consola).

## Comparación REST vs SOAP

REST y SOAP representan dos estilos de servicios web que pueden convivir dentro de una misma arquitectura SOA:

- REST es más ligero, usa JSON y se adapta bien a aplicaciones web y móviles.
- SOAP usa XML y WSDL, ofrece un contrato formal y es común en integraciones empresariales.
- Ambos pueden consumir la misma base de datos si comparten el mismo modelo de datos.
- Postman facilita las pruebas tanto de servicios REST como de SOAP.

Esta práctica mostró cómo se pueden combinar ambos enfoques para exponer la información académica de alumnos, calificaciones y matrículas.

## Conclusion

La actividad permitió aplicar conceptos teóricos de SOA en un caso práctico completo. Se diseñó una base de datos relacional, se implementó una API REST en Spring Boot y un servicio SOAP en Python, ambos conectados a la misma información.

El ejercicio ayudó a comprender:
- Cómo diseñar endpoints REST y operaciones SOAP.
- Cómo manejar la conexión a MySQL desde diferentes lenguajes.
- Cómo documentar y probar servicios con Postman.

En general, la práctica demuestra que SOA no depende de una sola tecnología, sino de la capacidad de exponer servicios bien definidos que puedan ser reutilizados por distintos clientes.

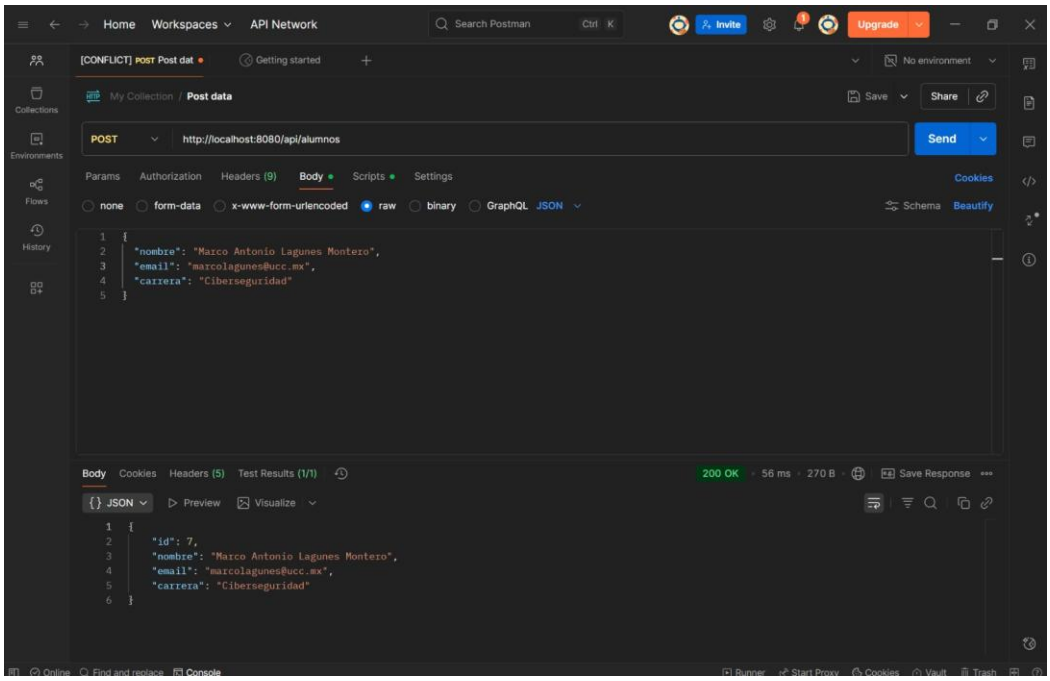
## Repositorio del proyecto en GitHub

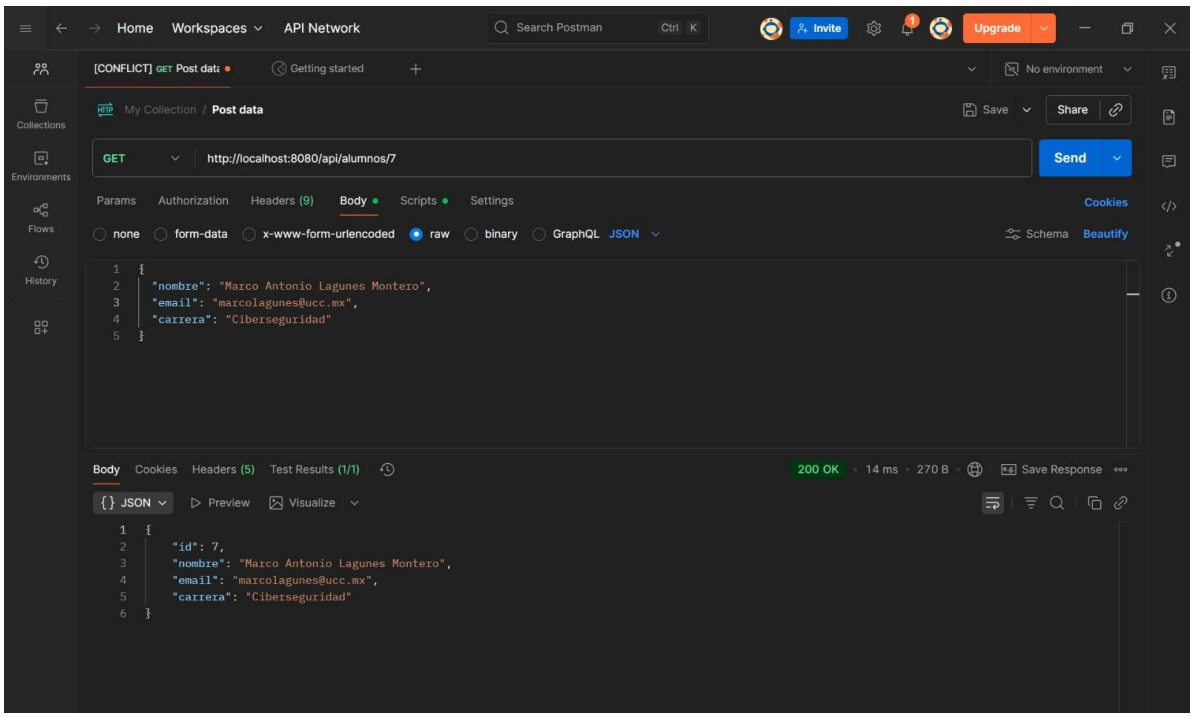
El código fuente del proyecto (API REST en Spring Boot, servicio SOAP en Python y scripts SQL) está disponible en el repositorio:

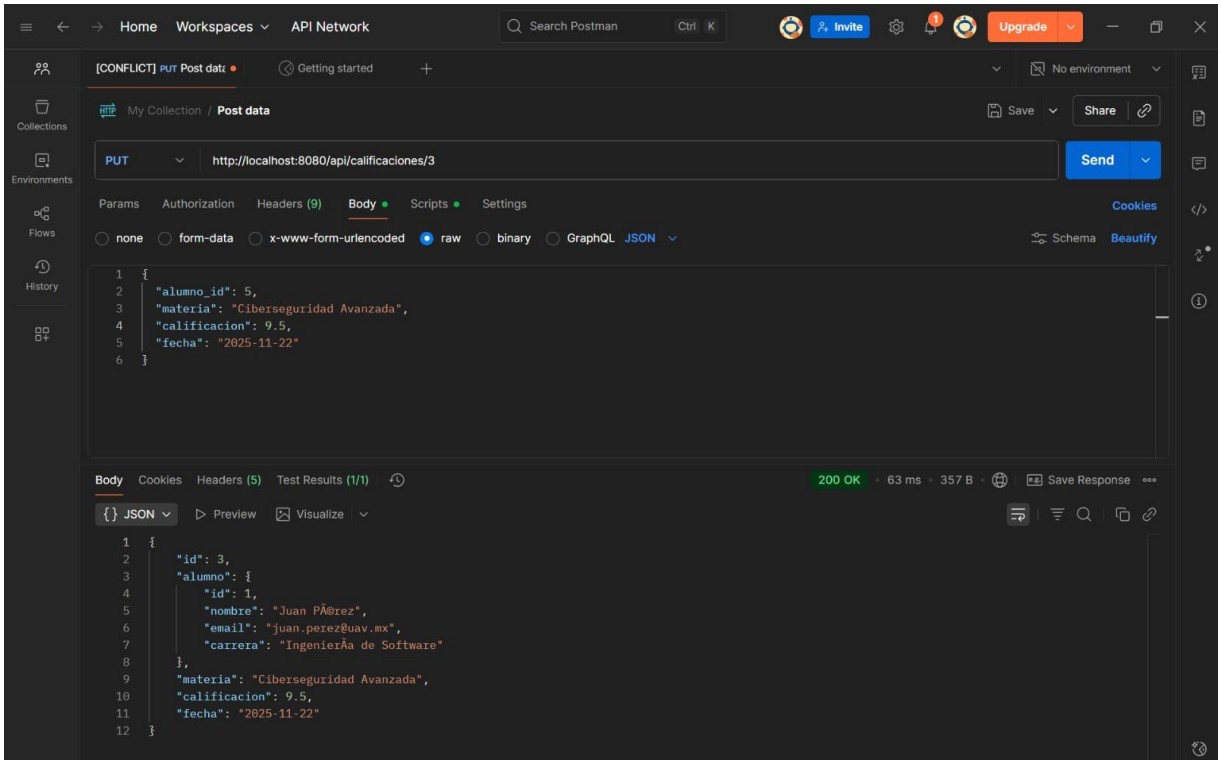
https://github.com/MarcoAntonioLagunes/RepoSOA2025

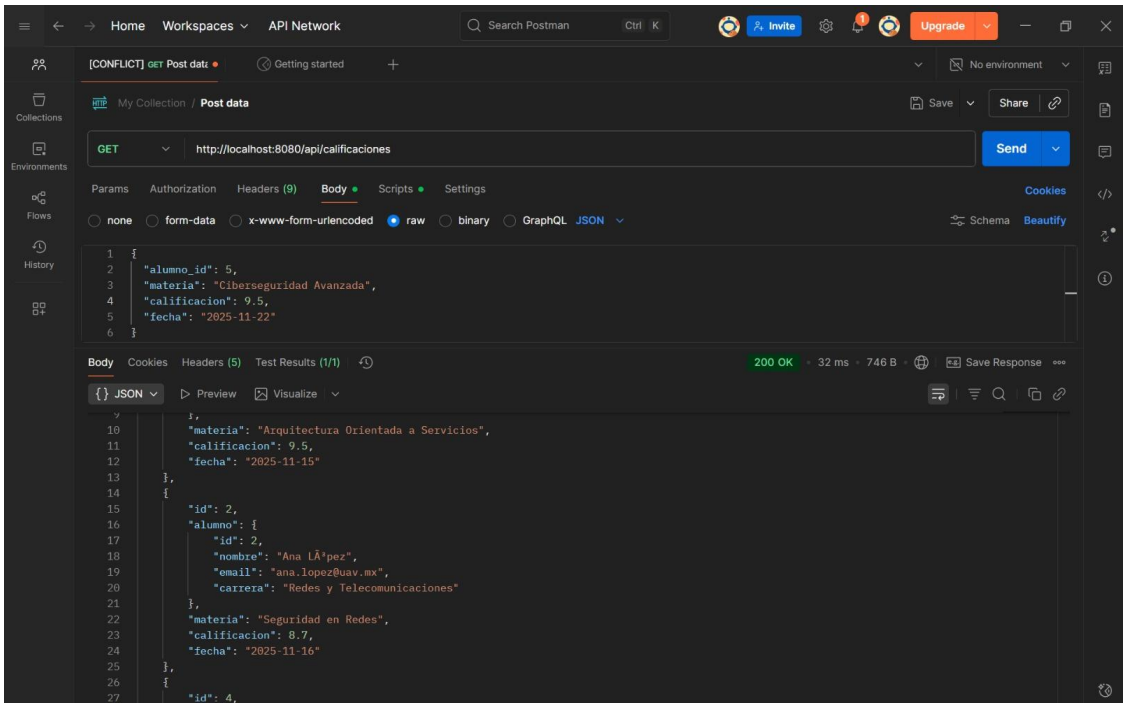
# Bibliografía

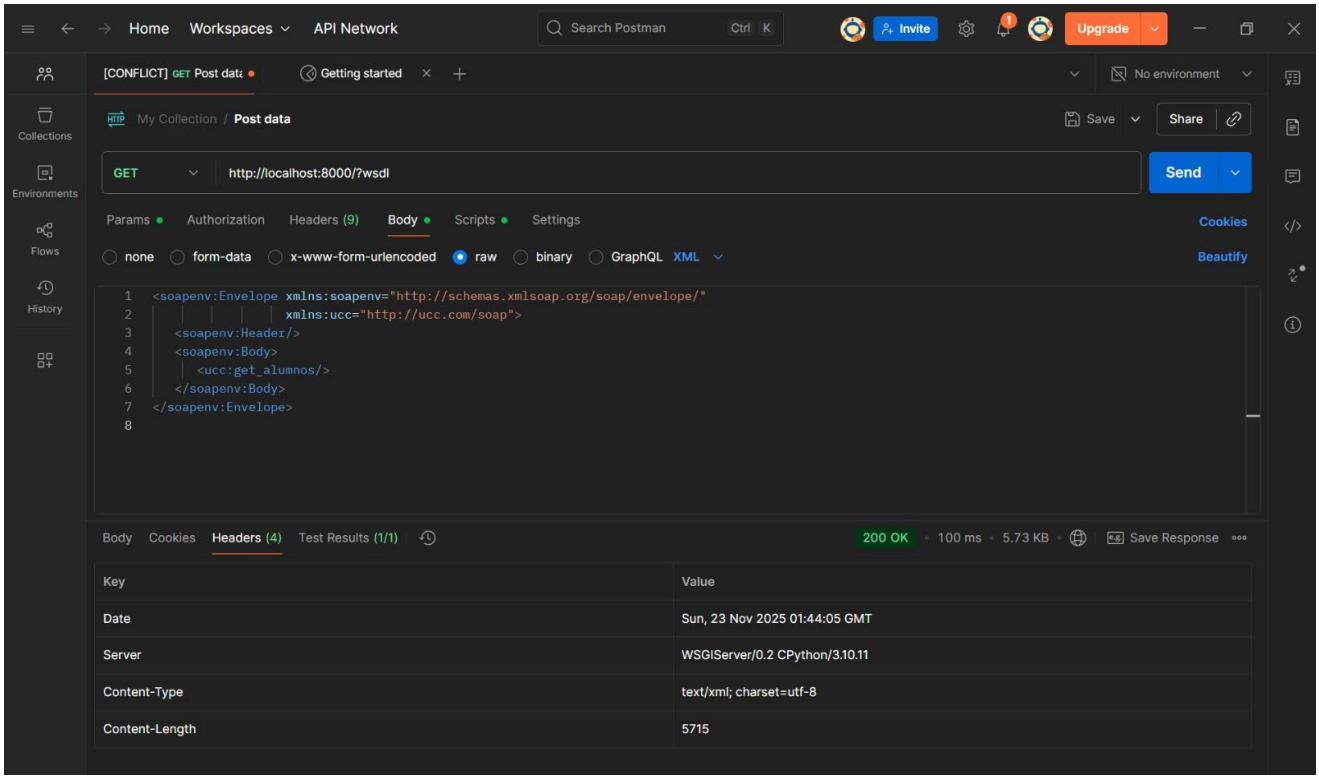

• Documentación oficial de Spring Boot – https://spring.io/projects/spring-boot
• Documentación de Spyne – https://spyne.io
• Notas y material proporcionado por el profesor Leonardo en clase.

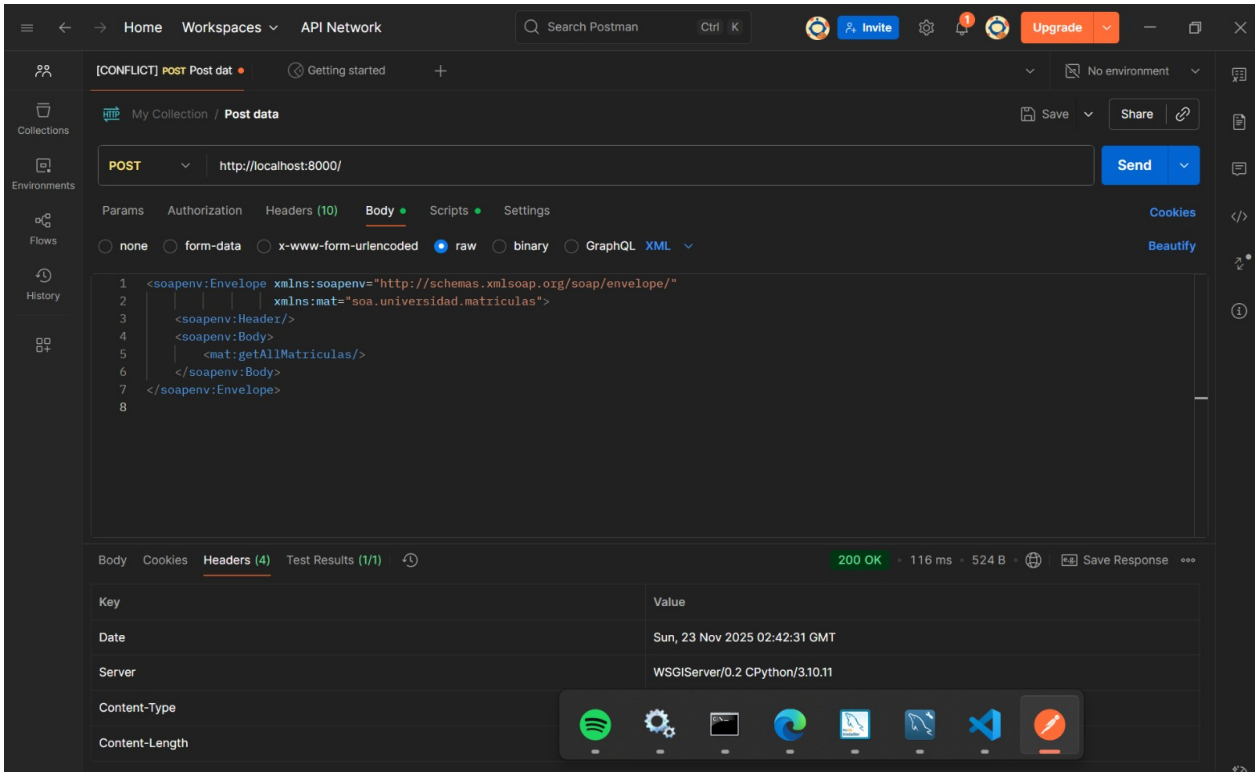



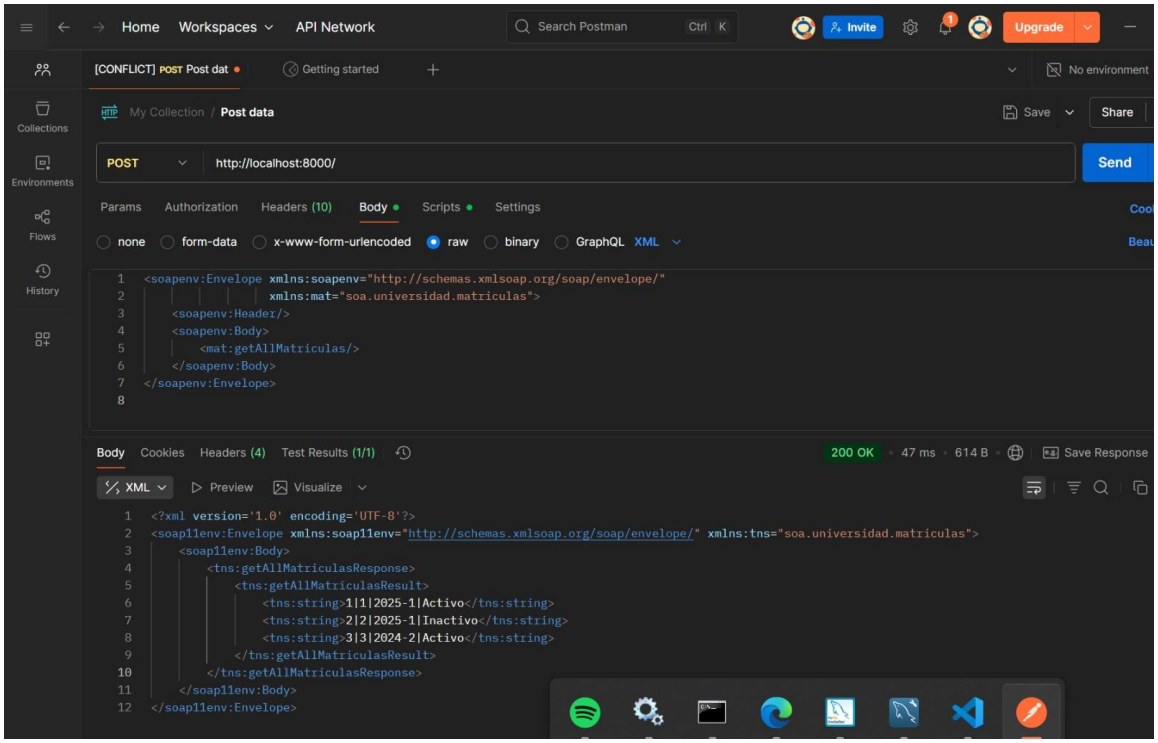



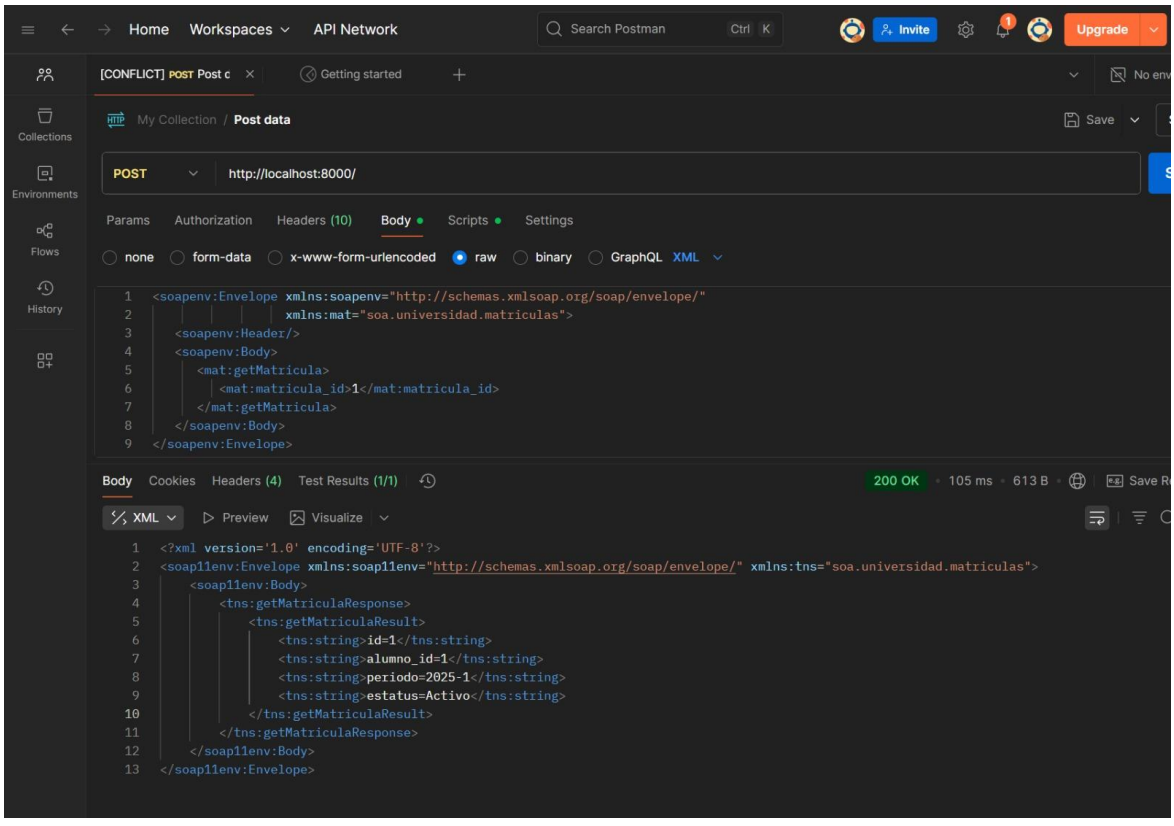

Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
[CONFLICT] PUT Post data
Getting started
No environment
Collections
Environments
Flows
History
My Collection / Post data
Save
Share
PUT
http://localhost:8080/api/calificaciones/3
Send
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL JSON
Schema
Beautify
{
"alumno_id": 5,
"materia": "Ciberseguridad Avanzada",
"calificacion": 9.5,
"fecha": "2025-11-22"
}
Body
Cookies
Headers (5)
Test Results (1/1)
200 OK
63 ms
357 B
Save Response
{ } JSON
Preview
Visualize
{
"id": 3,
"alumno": {
"id": 1,
"nombre": "Juan PÃ©rez",
"email": "juan.perez@uav.mx",
"carrera": "IngenierÃa de Software"
},
"materia": "Ciberseguridad Avanzada",
"calificacion": 9.5,
"fecha": "2025-11-22"
}


Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
[CONFLICT] GET Post data
Getting started
No environment
Collections
Environments
Flows
History
My Collection / Post data
Save
Share
GET
http://localhost:8080/api/calificaciones
Send
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL JSON
Schema
Beautify
{
"alumno_id": 5,
"materia": "Ciberseguridad Avanzada",
"calificacion": 9.5,
"fecha": "2025-11-22"
}
Body
Cookies
Headers (5)
Test Results (1/1)
200 OK
32 ms
746 B
Save Response
{ } JSON
Preview
Visualize
},
"materia": "Arquitectura Orientada a Servicios",
"calificacion": 9.5,
"fecha": "2025-11-15"
},
{
"id": 2,
"alumno": {
"id": 2,
"nombre": "Ana LÃ³pez",
"email": "ana.lopez@uav.mx",
"carrera": "Redes y Telecomunicaciones"
},
"materia": "Seguridad en Redes",
"calificacion": 8.7,
"fecha": "2025-11-16"
},
{
"id": 4,


Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
[CONFLICT] GET Post data
Getting started
No environment
Collections
Environments
Flows
History
My Collection / Post data
Save
Share
GET
http://localhost:8000/?wsdl
Send
Params
Authorization
Headers (9)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
XML
Beautify
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
2 xmlns:ucc="http://ucc.com/soap">
3 <soapenv:Header/>
4 <soapenv:Body>
5 <ucc:get_alumnos/>
6 </soapenv:Body>
7 </soapenv:Envelope>
8
Body
Cookies
Headers (4)
Test Results (1/1)
200 OK
100 ms
5.73 KB
Save Response
Key
Value
Date
Sun, 23 Nov 2025 01:44:05 GMT
Server
WSGIServer/0.2 CPython/3.10.11
Content-Type
text/xml; charset=utf-8
Content-Length
5715


Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
[CONFLICT] POST Post dat
Getting started
No environment
Collections
Environments
Flows
History
My Collection / Post data
Save
Share
POST
http://localhost:8000/
Send
Params
Authorization
Headers (10)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
XML
Beautify
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
2 xmlns:mat="soa.universidad.matriculas">
3 <soapenv:Header/>
4 <soapenv:Body>
5 <mat:getAllMatriculas/>
6 </soapenv:Body>
7 </soapenv:Envelope>
8
Body
Cookies
Headers (4)
Test Results (1/1)
200 OK
116 ms
524 B
Save Response
Key
Value
Date
Sun, 23 Nov 2025 02:42:31 GMT
Server
WSGIServer/0.2 CPython/3.10.11
Content-Type
Content-Length


Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
[CONFLICT] POST Post dat
Getting started
No environment
Collections
Environments
Flows
History
My Collection / Post data
Save
Share
POST
http://localhost:8000/
Send
Params
Authorization
Headers (10)
Body
Scripts
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL XML
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mat="soa.universidad.matriculas">
<soapenv:Header/>
<soapenv:Body>
<mat:getAllMatriculas/>
</soapenv:Body>
</soapenv:Envelope>
Body
Cookies
Headers (4)
Test Results (1/1)
200 OK
47 ms
614 B
Save Response
XML
Preview
Visualize
<?xml version='1.0' encoding='UTF-8'?>
<soap11env:Envelope xmlns:soap11env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tns="soa.universidad.matriculas">
<soap11env:Body>
<tns:getAllMatriculasResponse>
<tns:getAllMatriculasResult>
<tns:string>1|1|2025-1|Activo</tns:string>
<tns:string>2|2|2025-1|Inactivo</tns:string>
<tns:string>3|3|2024-2|Activo</tns:string>
</tns:getAllMatriculasResult>
</tns:getAllMatriculasResponse>
</soap11env:Body>
</soap11env:Envelope>


Home
Workspaces
API Network
Search Postman
Ctrl K
Invite
Upgrade
[CONFLICT] POST Post c
Getting started
Collections
Environments
Flows
History
My Collection / Post data
Save
POST
http://localhost:8000/
Params
Authorization
Headers (10)
Body
Scripts
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL XML
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mat="soa.universidad.matriculas">
<soapenv:Header/>
<soapenv:Body>
<mat:getMatricula>
<mat:matricula_id>1</mat:matricula_id>
</mat:getMatricula>
</soapenv:Body>
</soapenv:Envelope>
Body
Cookies
Headers (4)
Test Results (1/1)
200 OK
105 ms
613 B
XML
Preview
Visualize
<?xml version='1.0' encoding='UTF-8'?>
<soap11env:Envelope xmlns:soap11env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tns="soa.universidad.matriculas">
<soap11env:Body>
<tns:getMatriculaResponse>
<tns:getMatriculaResult>
<tns:string>id=1</tns:string>
<tns:string>alumno_id=1</tns:string>
<tns:string>periodo=2025-1</tns:string>
<tns:string>estatus=Activo</tns:string>
</tns:getMatriculaResult>
</tns:getMatriculaResponse>
</soap11env:Body>
</soap11env:Envelope>