

Universidad Cristóbal Colón

Arquitectura Orientada a Servicios (SOA)

**Reporte de pruebas con POSTMAN
APIs REST y SOAP**

Proyecto: Actividad de clase 21/11/2025

Nombre de la alumna: Isis Villanueva
Carrera: Ingeniería en Sistemas Computacionales
Profesor: Mtro. Leonardo

Fecha: 21 de noviembre de 2025

1. Introducción

El presente documento describe las pruebas funcionales realizadas con POSTMAN sobre dos servicios desarrollados para la plataforma académica basada en SOA:

a) una API REST implementada en Java con Spring Boot y MySQL, encargada de la gestión de alumnos y calificaciones; y b) un servicio SOAP implementado en Python con Spyne y MySQL, responsable de la gestión de matrículas.

El objetivo de las pruebas es verificar que los endpoints expuestos por ambas APIs responden correctamente ante operaciones de consulta, alta y validación de datos, cumpliendo con los requerimientos de interoperabilidad definidos en la actividad.

2. Descripción general de las APIs probadas

API REST (Java + Spring Boot):

- Entidad principal: Alumno, asociada a la entidad Calificación.
- Base de datos: MySQL (schema soa_universidad).
- Principales endpoints probados:
 - GET /alumnos
 - GET /alumnos/{id}
 - POST /alumnos
 - GET /calificaciones
 - GET /calificaciones/{id}

API SOAP (Python + Spyne):

- Servicio: MatriculaService.
- Operaciones principales:
 - getAllMatriculas(): devuelve todas las matrículas registradas.
 - getMatricula(matricula_id): consulta una matrícula por ID.
 - createMatricula(alumno_id, periodo, estatus): registra una nueva matrícula.

3. Ambiente y herramientas de prueba

- Cliente de pruebas: POSTMAN.
- Motor de base de datos: MySQL local.
- API REST: proyecto Java Spring Boot ejecutándose en <http://localhost:8080>.
- API SOAP: servicio Python Spyne ejecutándose en <http://localhost:8000> con WSDL expuesto en <http://localhost:8000/?wsdl>.
- Sistema operativo: Windows 11.

4. Casos de prueba para la API REST (Java)

Caso	Endpoint	Método	Datos de entrada	Resultado esperado	Resultado obtenido
R1	GET /alumnos	GET	-	Listado completo de alumnos con código 200 OK	Se devuelve el listado de alumnos almacenados en MySQL con código 200 OK.
R2	GET /alumnos/1	GET	id = 1	Datos del alumno con ID=1 o mensaje de no encontrado	Se muestra la información del alumno 1 correctamente con código 200 OK.
R3	POST /alumnos	POST	{ "nombre": "Nuevo Alumno", "email": "nuevo@ucc.mx" }	Creación de un nuevo alumno con código 201 Created	El endpoint registra el alumno en la tabla alumnos y responde 201 Created.
R4	GET /calificaciones	GET	-	Listado de calificaciones relacionadas con alumnos existentes	Se obtienen todas las calificaciones registradas, incluyendo las de prueba cargadas por script SQL.

En todos los casos anteriores, los datos obtenidos coinciden con el contenido de la base de datos MySQL, validando la correcta integración entre Spring Boot y el motor de base de datos.

5. Casos de prueba para la API SOAP (Python)

Caso	Operación	SOAP Action / Body	Datos de entrada	Resultado esperado	Resultado obtenido
------	-----------	--------------------	------------------	--------------------	--------------------

o

S1	getAllMatriculas	POST http://localhost:8000/	Envelope con la operación mat:getAllMatriculas	Lista de todas las matrículas registradas en formato XML dentro del cuerpo SOAP.	Se reciben varias entradas en <tns:string> con el formato id alumno_id periodo estatus.
S2	getMatricula	POST http://localhost:8000/	Envelope con mat:matricula_id=1	Datos de la matrícula a con ID=1 si existe.	El servicio responde un conjunto de elementos <tns:string> con los atributos de la matrícula 1.
S3	createMatricula	POST http://localhost:8000/	Envelope con alumno_id=5, periodo='2025-1', estatus='Activo'	Registro exitoso de una nueva matrícula asociada al alumno 5.	El servicio inserta el registro en la tabla matriculas y responde con el mensaje 'Matricula creada correctamente'.

6. Evidencias de ejecución en POSTMAN

Durante las pruebas se capturaron evidencias gráficas donde se aprecia:

- La petición POST al servicio SOAP con el envelope XML y la respuesta con la lista de matrículas.
- Las peticiones GET y POST a la API REST, mostrando el cuerpo JSON de respuesta y los códigos 200/201 en la sección de status de POSTMAN.

Estas evidencias confirman que la configuración de cabeceras (Content-Type text/xml para SOAP y application/json para REST) es correcta y que los endpoints están disponibles desde el cliente.

7. Conclusiones

Las pruebas realizadas con POSTMAN permiten concluir que la API REST en Java y el servicio SOAP en Python cumplen con los requerimientos mínimos de la actividad:

- Exponen operaciones claras para gestionar información académica (alumnos, calificaciones y matrículas).
- Responden correctamente ante peticiones válidas devolviendo códigos HTTP adecuados.
- Mantienen la coherencia de datos con la base de datos MySQL.

Como trabajo futuro se propone automatizar estos casos de prueba con colecciones de POSTMAN y scripts de prueba, así como integrar ambas APIs detrás de un gateway unificado para facilitar el consumo desde aplicaciones web o móviles.