

# Desarrollo de una API RESTful

## Objetivos:

Que desarrollen una API RESTful utilizando Python y para gestionar una base de datos y proporcionar un servicio accesible por otros sistemas.

Que demuestren sus conocimientos sobre desarrollo de APIs, manejo de bases de datos, seguridad, autenticación, y buenas prácticas de desarrollo en Python.

## Caso de uso ejemplo:

Desarrollar una API para un sistema de gestión de reservas de un gimnasio.

Los casos de uso son:

- RQ1 - Los usuarios podrán registrarse
- RQ2 - Los usuarios podrán ver horarios
- RQ3 - Los usuarios podrán reservar clases
- RQ4 - Los usuarios podrán cancelar reservas
- RQ5 - Los usuarios podrán recibir notificaciones ante cambios de fechas/horarios

## Consigna:

Cada alumno/grupo deberá plantear un caso de uso similar al planteado arriba, indicando al menos 5 requerimientos de alto nivel, los cuales deberán estar contemplados en los endpoints de la API.

## Stack mínimo:

1. Framework para la API
2. Autenticación: para la autenticación segura de los usuarios.
3. Base de datos
4. Documentación: Generación automática de documentación
5. Contenedores: para contenerizar la aplicación y facilitar su despliegue.
6. Testing
7. Deploy

## Requisitos adicionales:

- La API debe seguir principios RESTful.
- Deben implementarse mecanismos de validación y manejo de errores.
- Se espera que los estudiantes entreguen un código bien documentado y con una correcta separación de responsabilidades.
- Aplicar, al menos, 4 patrones de diseño de los vistos que tengan sentido y utilidad real.
- Incluir al menos 10 tests unitarios que validan el funcionamiento de la API.
- Proveer un archivo README con instrucciones claras sobre cómo desplegar y probar la API (con Postman). Incluir JSON para pruebas unitarias.
- Incluir una presentación indicando:
  - Al menos 1 diagrama UML
  - Dónde, porque y qué patrones de diseño se utilizaron

## Evaluación:

1. Funcionalidad (20%): Si la API cumple correctamente con los requerimientos funcionales y maneja correctamente los errores.
2. Calidad del código (20%): Claridad de código y uso adecuado de conceptos vistos en la materia.
3. Documentación y pruebas (20%): Calidad de la documentación y cobertura de las pruebas.
4. Exposición y defensa (40%): claridad de conceptos y justificación de los conceptos aplicados.