



## LABORATORIO NO. 06

### Creación de APIs para Pruebas

---

#### CASO A:

Desarrollar una API RESTful para gestionar el registro de usuarios en una plataforma.

#### Requisitos Funcionales:

##### 1. Creación de Usuario (POST /usuarios):

- Se debe permitir el registro de un nuevo usuario con los siguientes campos:
  - **Nombre completo:** Obligatorio. Solo se aceptan caracteres alfabéticos. Mínimo 2 palabras.
  - **Correo electrónico:** Obligatorio. Debe ser un correo electrónico válido y único (no se permiten duplicados).
  - **Contraseña:** Obligatorio. Debe tener al menos 8 caracteres, incluyendo una mayúscula, una minúscula, un número y un carácter especial.
  - **Edad:** Opcional. Si se proporciona, debe ser un número entre 18 y 120.
  - **País:** Obligatorio. Debe ser un país válido (puedes usar una lista de países predefinida).
  - **Número de teléfono:** Opcional. Debe cumplir con un formato específico de número internacional (Ej. +1-123-456-7890).

##### 2. Lectura de Usuarios (GET /usuarios):

- Se debe permitir la consulta de todos los usuarios registrados, mostrando los datos en formato JSON.

##### 3. Actualización de Usuario (PUT /usuarios/{id}):

- Permitir la actualización de los datos de un usuario específico. Todas las validaciones mencionadas en la creación deben aplicarse al momento de la actualización.

##### 4. Eliminación de Usuario (DELETE /usuarios/{id}):

- Permitir la eliminación de un usuario específico por su ID.



## CASO B:

Desarrollar una API RESTful para gestionar un catálogo de productos.

### Requisitos Funcionales:

#### 1. Creación de Producto (POST /productos):

- Se debe permitir la creación de un nuevo producto con los siguientes campos:
  - **Nombre del producto:** Obligatorio. Debe tener entre 3 y 100 caracteres.
  - **Descripción:** Opcional. No más de 500 caracteres.
  - **Precio:** Obligatorio. Debe ser un número positivo mayor a 0 y con un máximo de dos decimales.
  - **Categoría:** Obligatorio. Debe seleccionarse de una lista predefinida de categorías (como Electrónica, Ropa, Alimentos, etc.).
  - **Inventario:** Obligatorio. Debe ser un número entero no negativo (cantidad disponible en stock).
  - **SKU (Código único del producto):** Obligatorio. Debe ser único, solo puede contener letras, números y guiones (-), y debe tener entre 6 y 20 caracteres.
  - **Fecha de lanzamiento:** Opcional. Debe ser una fecha válida en el futuro o el presente.
  - **Imagen del producto (URL):** Opcional. Si se proporciona, debe ser una URL válida.

#### 2. Lectura de Productos (GET /productos):

- Se debe permitir la consulta de todos los productos en el catálogo, retornando la información en formato JSON.
- Debe permitirse la búsqueda de productos por nombre o categoría y filtrar los resultados por rango de precios.

#### 3. Lectura de un Producto Específico (GET /productos/{id}):

- Se debe permitir la consulta de un producto específico usando su ID o SKU.

#### 4. Actualización de Producto (PUT /productos/{id}):

- Permitir la actualización de los detalles de un producto específico. Todas las validaciones mencionadas en la creación de productos deben aplicarse durante la actualización.

#### 5. Eliminación de Producto (DELETE /productos/{id}):

- Permitir la eliminación de un producto específico mediante su ID.



## CASO C:

Desarrollar una API RESTful para la gestión de clientes como parte de un sistema CRM.

### Requisitos Funcionales:

#### 1. Creación de Cliente (POST /clientes):

- Se debe permitir registrar un nuevo cliente con los siguientes campos:
  - **Nombre completo:** Obligatorio. Debe tener entre 3 y 100 caracteres, solo se aceptan letras y espacios.
  - **Correo electrónico:** Obligatorio. Debe ser un correo electrónico válido y único.
  - **Teléfono:** Obligatorio. Debe ser un número en formato internacional (Ej: +1-123-456-7890).
  - **Empresa:** Opcional. Nombre de la empresa con la que está asociado el cliente, hasta 100 caracteres.
  - **País:** Obligatorio. Validar que el país esté en una lista predefinida.
  - **Fecha de contacto:** Obligatorio. Debe ser una fecha válida en el pasado o el presente (fecha de cuando el cliente fue contactado por primera vez).
  - **Estado del cliente:** Obligatorio. Debe ser uno de los siguientes valores predefinidos: "Nuevo", "En negociación", "Ganado", "Perdido".

#### 2. Lectura de Clientes (GET /clientes):

- Se debe permitir la consulta de todos los clientes registrados, retornando los datos en formato JSON.
- Debe permitirse filtrar clientes por:
  - **Estado del cliente** (Nuevo, En negociación, Ganado, Perdido).
  - **Empresa.**
  - **Rango de fechas de contacto.**

#### 3. Lectura de Cliente Específico (GET /clientes/{id}):

- Se debe permitir la consulta de un cliente específico por su ID.

#### 4. Actualización de Cliente (PUT /clientes/{id}):



- Permitir la actualización de los datos de un cliente específico. Todas las validaciones mencionadas en la creación de clientes deben aplicarse en la actualización.
- 5. **Eliminación de Cliente (DELETE /clientes/{id}):**
  - Permitir la eliminación de un cliente específico mediante su ID.

#### CASO D:

Desarrollar una API RESTful para gestionar tareas o procesos dentro de una empresa.

#### Requisitos Funcionales:

##### 1. Creación de Tarea (POST /tareas):

- Se debe permitir registrar una nueva tarea o proceso con los siguientes campos:
  - **Título de la tarea:** Obligatorio. Debe tener entre 5 y 100 caracteres.
  - **Descripción:** Opcional. No más de 500 caracteres, donde se detalle el propósito o los pasos de la tarea.
  - **Fecha de creación:** Obligatorio. Debe ser la fecha en la que se crea la tarea, y no puede ser una fecha futura.
  - **Fecha límite:** Opcional. Si se proporciona, debe ser una fecha válida en el futuro.
  - **Prioridad:** Obligatorio. Debe ser uno de los siguientes valores: "Baja", "Media", "Alta".
  - **Estado de la tarea:** Obligatorio. Inicialmente debe ser "Pendiente". Los posibles estados son: "Pendiente", "En Proceso", "Completada", "Cancelada".
  - **Usuario asignado:** Opcional. Debe ser un identificador único (ID) de un empleado de la empresa. Si se proporciona, debe corresponder a un usuario existente.
  - **Categoría:** Obligatorio. Debe seleccionarse de una lista predefinida de categorías (Ej. "Administración", "Ventas", "Soporte", "Desarrollo").

##### 2. Lectura de Tareas (GET /tareas):

- Se debe permitir la consulta de todas las tareas registradas, retornando la información en formato JSON.
- Debe permitirse filtrar tareas por:



- **Estado de la tarea** (Pendiente, En Proceso, Completada, Cancelada).
  - **Prioridad** (Baja, Media, Alta).
  - **Fecha límite**: Buscar las tareas que deben completarse en un rango de fechas.
  - **Usuario asignado**: Filtrar por el ID del empleado asignado.
3. **Lectura de una Tarea Específica (GET /tareas/{id})**:
- Se debe permitir la consulta de una tarea específica usando su ID.
4. **Actualización de Tarea (PUT /tareas/{id})**:
- Permitir la actualización de los datos de una tarea específica. Todas las validaciones mencionadas en la creación de tareas deben aplicarse durante la actualización.
  - Se debe permitir cambiar el **estado** de la tarea, con las siguientes restricciones:
    - Solo se puede pasar a "Completada" si el estado actual es "En Proceso".
    - Una vez que una tarea está en estado "Completada" o "Cancelada", no se puede modificar de nuevo.
5. **Eliminación de Tarea (DELETE /tareas/{id})**:
- Permitir la eliminación de una tarea específica por su ID. No se pueden eliminar tareas que ya estén en estado "Completada" o "Cancelada".