

TALLER 3: Inventario de Endpoints + API en Node/TypeScript (sin persistencia)

Endpoints						
<div>GET POST DELETE/PUT/PATCH</div>						
Método	Ruta	Params	Body (Request)	Response	Descripción	Notas de Seguridad
GET	/users	page?, limit?, search?	—	<div>200: {success: true, data: Usuario[], timestamp}</div> <div>500: error interno</div>	Obtiene lista paginada de usuarios. Soporta búsqueda por nombre/apellido. Validaciones para parámetros opcionales.	No
GET	/users/:id	id (path)	—	<div>200: {success: true, data: Usuario}</div> <div>404 / 400</div>	Obtiene un usuario específico por ID. Validaciones: ID alfanumérico.	No
POST	/users	—	{nombre, apellido, edad, email, telefono}	<div>201: {success: true, data: Usuario}</div> <div>400: VALIDATION_ERROR</div>	Crea un nuevo usuario. Validaciones: campos requeridos, email válido, edad 0–150.	No
PUT	/users/:id	id (path)	{nombre, apellido, edad, email, telefono}	<div>200: {success: true, data: Usuario}</div> <div>404 / 400</div>	Actualiza completamente un usuario. Todos los campos requeridos.	No
PATCH	/users/:id	id (path)	{nombre?, apellido?, edad?, email?, telefono?}	<div>200: {success: true, data: Usuario}</div> <div>404 / 400</div>	Actualiza parcialmente los campos enviados. Al menos 1 campo requerido.	No
DELETE	/users/:id	id (path)	—	<div>200: {success: true, data: null}</div> <div>404</div>	Elimina un usuario por ID.	No
GET	/users/stats	—	—	<div>200: {success: true, data: {total, averageAge, totalByAgeRange}}</div>	Obtiene estadísticas de usuarios: total, edad promedio y distribución por rangos.	No
GET	/users/search	q, limit?	—	<div>200: {success: true, data: Usuario[]}</div> <div>400: VALIDATION_ERROR</div>	Busca usuarios por nombre, apellido o email. Param q requerido.	No

[link a postman](#) clic ahí