

# Casos de Prueba

## Verificación de Datos Iniciales

Tabla **Estudiantes** :

id	nombre	carrera
1	Alejandro Ramírez	Ingeniería de Software
2	Laura Méndez	Administración de Empresas
3	Felipe Torres	Ingeniería de Software
4	Diana López	Contaduría Pública
5	Camila Suárez	Administración de Empresas
6	Santiago Vega	Ingeniería de Software

Tabla **Materias** :

id	nombre	creditos	profesor_id
1	Matemáticas Avanzadas	3	1
2	Física Cuántica	3	1
3	Programación en Python	3	2
4	Bases de Datos	3	2
5	Inteligencia Artificial	3	3
6	Redes de Computadoras	3	3
7	Ética Profesional	3	4
8	Gestión de Proyectos	3	4
9	Marketing Digital	3	5
10	Finanzas Empresariales	3	5
11	Ciencia de Datos	3	NULL

Tabla **Registros** :

id	estudiante_id	materia_id
1	1	1
2	1	3
3	1	5
4	2	2
5	2	4
6	2	6
7	3	7
8	3	8
9	3	9
10	4	1
11	4	5
12	4	10
13	5	3
14	5	6
15	5	7

## Endpoints de la API

Entidad	Método	Endpoint	Descripción
Estudiantes	POST	/estudiantes	Crear un nuevo estudiante.
Estudiantes	GET	/estudiantes	Obtener la lista de todos los estudiantes.
Estudiantes	GET	/estudiantes/{id}	Obtener un estudiante por su ID.
Estudiantes	PUT	/estudiantes/{id}	Actualizar un estudiante por su ID.
Estudiantes	DELETE	/estudiantes/{id}	Eliminar un estudiante por su ID.
Estudiantes	GET	/estudiantes/{id}/compañeros	Obtener los nombres de los compañeros con los

Entidad	Método	Endpoint	Descripción
			que un estudiante comparte cada materia.
Estudiantes	GET	/estudiantes/{id}/visibilidad	Obtener la lista de todos los estudiantes registrados, mostrando solo los nombres con los cuales se comparte materias.
Registros	POST	/registros	Registrar a un estudiante en materias cumpliendo las reglas de selección.
Materias	GET	/materias	Obtener la lista de materias disponibles.
Materias	GET	/materias/{id}	Obtener información de una materia específica.
Profesores	GET	/profesores	Obtener la lista de todos los profesores.
Profesores	GET	/profesores/{id}	Obtener un profesor por su ID.

# Batería de Pruebas - API

## 1. Estudiantes

### Crear Estudiante (POST)

Tipo	Entrada	Salida	Código
Caso de éxito	{ "nombre": "Juan Pérez", "carrera": "Ingeniería de Software" }	{ "id": 7, "nombre": "Juan Pérez", "carrera": "Ingeniería de Software" }	201 Created
Caso de error (nombre inválido)	{ "nombre": "Jo", "carrera": "Ingeniería de Software" }	{ "error": "El nombre debe tener al menos 3 caracteres" }	400 Bad Request

### Listar Todos (GET)

Tipo	Entrada	Salida	Código
Caso de éxito	Ninguna	[{ "id": 1, "nombre": "Juan Pérez", "carrera": "Ingeniería de Software" }]	200 OK

## Buscar por ID (GET)

Tipo	Entrada	Salida	Código
Caso de éxito	ID: 1	{ "id": 1, "nombre": "Juan Pérez", "carrera": "Ingeniería de Software" }	200 OK
Caso de error (no encontrado)	ID: 999	{ "error": "Estudiante no encontrado" }	404 Not Found

## Editar Estudiante (PUT)

Tipo	ID	Entrada	Salida	Código
Caso de éxito	ID: 1	{ "nombre": "Nuevo Nombre", "carrera": "Ingeniería de Software" }	{ "id": 1, "nombre": "Nuevo Nombre", "carrera": "Ingeniería de Software" }	200 OK
Caso de error (no encontrado)	ID: 999	{ "nombre": "Nuevo Nombre", "carrera": "Ingeniería de Software" }	{ "error": "Estudiante no encontrado" }	404 Not Found

## Eliminar Estudiante (DELETE)

Tipo	Entrada	Salida	Código
Caso de éxito	ID: 7	{ "mensaje": "Eliminado correctamente" }	200 OK
Caso de error (con materias)	ID: 1	{ "error": "No se puede eliminar un estudiante con materias inscritas" }	400 Bad Request

Tipo	Entrada	Salida	Código
Caso de error (no encontrado)	ID: 999	{ "error": "No encontrado" }	404 Not Found

## Ver Compañeros de Clase (GET)

Tipo	ID	Entrada	Salida	Código
Caso de éxito	1	Ninguna	[{"NombreEstudiante": "Diana López", "NombreMateria": "Matemáticas Avanzadas"}]	200 OK
Caso de error (sin materias)	7	Ninguna	{ "error": "No tiene materias asignadas" }	400 Bad Request
Caso de error (no existe el estudiante)	999	Ninguna	{"error": "No se encontró el ID del estudiante."}	400 Bad Request

## Ver Visibilidad de Otros Estudiantes (GET)

Tipo	Entrada	Salida	Código
Caso de éxito	ID: 1	[{"estudiante": "Diana López", "materia": "Matemáticas Avanzadas", "profesor": "Carlos Pérez"}]	200 OK
Caso de error (estudiante no encontrado)	ID: 999	{ "error": "No se encontraron resultados para el estudiante especificado" }	404 Not Found
Caso de error (ID inválido)	ID: 0	{ "error": "ID de estudiante no válido" }	400 Bad Request

## 2. Registros

### Inscripción de Materias (POST)

Tipo	Entrada	Salida	Código
Caso de éxito	{ "estudiante_id": 1, "materia_id": 1 }	{ "id": 1, "estudiante_id": 1, "materia_id": 1 }	201 Created

Tipo	Entrada	Salida	Código
Caso de error (ya inscrito)	{ "estudiante_id": 1, "materia_id": 1 }	{ "error": "El estudiante ya está inscrito en esta materia" }	400 Bad Request
Caso de error (máximo de materias)	{ "estudiante_id": 1, "materia_id": 2 }	{ "error": "Máximo 3 materias por estudiante" }	400 Bad Request
Caso de error (mismo profesor)	{ "estudiante_id": 1, "materia_id": 2 }	{ "error": "No puedes tener dos materias con el mismo profesor" }	400 Bad Request
Caso de error (límite de estudiantes)	{ "estudiante_id": 2, "materia_id": 1 }	{ "error": "La materia ha alcanzado el límite de estudiantes" }	400 Bad Request

### 3. Materias

#### Listar Todas las Materias (GET)

Tipo	Entrada	Salida	Código
Caso de éxito	Ninguna	[{ "id": 1, "nombre": "Matemáticas Avanzadas", "creditos": 3, "profesor_id": 1 }]	200 OK

### 4. Profesores

#### Listar Todos los Profesores (GET)

Tipo	Entrada	Salida	Código
Caso de éxito	Ninguna	[{ "id": 1, "nombre": "Juan Pérez" }]	200 OK

#### Buscar Profesor por ID (GET)

Tipo	Entrada	Salida	Código
Caso de éxito	ID: 1	{ "id": 1, "nombre": "Carlos Pérez", "materias": ["Matemáticas Avanzadas", "Física Cuántica"] }	200 OK

Tipo	Entrada	Salida	Código
Caso de error (no encontrado)	ID: 999	{ "error": "Profesor no encontrado" }	404 Not Found