

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE



**ITESO, Universidad
Jesuita de Guadalajara**

DESARROLLO DE APLICACIONES Y SERVICIOS WEB

Presenta:

Mónica Belén Sosa

Diego Esquivel

Carlos Moreno

Tlaquepaque, Jalisco a 26 de Noviembre del 2025

https://iteso01-my.sharepoint.com/:v/g/personal/carlos_moreno_iteso_mx/IQCpaQ772PJjR5p83RlX4BM9ASqCs2XgUxQjt5NV1wOB6cc?e=y7rrDv&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoIJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOijTaGFyZURpYWxvZy1MaW5rliwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYilsInJlZmVycmFsTW9kZSI6InZpZXcifX0%3D

METODO	RUTA(ENDPOINT)	DESCRIPCION	COMP LETAD O?	AUTH REQUERIDO?	NOTAS
POST	/api/auth/login	Inicio sesión	si	no	Retorna JWT
POST	/api/auth/logout	Cerrar sesión	si	si	Elimina el token
GET	/api/auth/login	Obtener usuario logeado	si	si	Obtener usuario logueado con token
GET	/api/usuarios	Obtener usuarios	si	si	Soporta ?page=1&limit=10 y filtros
POST	/api/usuarios	Crear usuario	si	si	Requiere rol admin
PUT	/api/usuarios/:id	Actualizar usuario	si	si	Requiere rol admin
DELETE	/api/usuarios/:id	Borrar usuario	si	si	Requiere rol admin
GET	/api/asignaturas	Obtener asignaturas	si	si	Soporta ?page=1&limit=10 y filtros
POST	/api/asignaturas	Crear asignatura	si	si	Requiere rol admin
PUT	/api/asignaturas/:id	Actualizar asignatura	si	si	Requiere rol admin
DELETE	/api/asignaturas/:id	Borrar asignatura	si	si	Requiere rol admin
GET	/api/evaluaciones	Obtener evaluaciones	si	si	Soporta ?page=1&limit=10 y filtros
POST	/api/evaluaciones	Crear evaluaciones	si	si	Requiere rol admin
PUT	/api/evaluaciones/:id	Actualizar evaluaciones	si	si	Requiere rol admin
DELETE	/api/evaluaciones/:id	Borrar evaluaciones	si	si	Requiere rol admin
GET	/api/profesores	Obtener profesores	si	si	Soporta ?page=1&limit=10 y filtros
POST	/api/profesores	Crear profesores	si	si	Requiere rol admin
PUT	/api/profesores/:id	Actualizar profesores	si	si	Requiere rol admin
DELETE	/api/profesores/:id	Borrar profesores	si	si	Requiere rol admin

Evidencia de filtros y paginación

```
export const obtenerAsignaturas = async (query) => {
  const { page = 1, limit = 10, nombre, codigo } = query;

  const filtros = {};
  if (nombre) filtros.nombre = new RegExp(nombre, "i");
  if (codigo) filtros.codigo = codigo.toUpperCase();

  const asignaturas = await Asignatura.find(filtros)
    .skip((page - 1) * limit)
    .limit(Number(limit));

  const total = await Asignatura.countDocuments(filtros);

  return { asignaturas, total };
};

export const crearAsignatura = async (data) => {
  const asignatura = await Asignatura.create(data);
  return asignatura;
};

export const actualizarAsignatura = async (codigo, data) => {
  const asignatura = await Asignatura.findByIdAndUpdate(codigo, data, {
    new: true,
  });

  return asignatura;
};
```

en ese pantallazo del get de asignaturas podemos ver cómo se implementan tanto los filtros como la paginación.

como funciona:

1. La función recibe query, en query extrae page (si no lo tiene asigna el 1), extrae limit (si no lo tiene asigna 10), extrae nombre (opcional y extrae código (opcional)
2. Los agrega a filtros
3. Busca en la base de datos con esos filtros .
4. Skip es para saltar entonces agarra page y lo multiplica por limit por ejemplo si page es 1 y limit es 10 seria $(1-1)*10=0$ por lo que se va a saltar 0 resultados pero si page fuera 2 seria $(2-1)*10=10$ por lo tanto se va a saltar los primeros 10 registros encontrados.
5. Limit es para limitar el numero de registros que te tiene que traer por ejemplo si se asigna 10 a limit te va a traer nada mas los 10 registros que encontró.

Con eso se explica como funciona tanto los filtros como la paginación.

distribución del trabajo

- **DIEGO:** profesores, evaluación .
- **MONICA:** Asignaturas.
- **CARLOS:** logueo, usuario.

Pendientes

Lo único que tenemos pendiente son desplegar el backend y el frontend para que el profesor pueda poder ingresar desde su computadora, unir el backend con el frontend para que el front consuma el back y seria todo ya que todo el back esta terminado.

Dificultades.

Lo que se dificulto un poco fue tantos endpoints que se tuvieron que hacer, se nos hizo muy tardado tantos endpoint, también un poco el tema de la paginación, saber como es que se realizaba