El encapsulamiento es uno de los principios fundamentales de la programación orientada a objetos, sin embargo también puede ser llevado a diferentes a otros estilos de programación.

Para entender el concepto debemos tener claro que el encapsulamiento es una herramienta para agrupar y controlar, esas dos cosas las podemos aplicar a un sin numero de otras, podemos agrupar y controlar, datos, funciones, clases, paquetes, servicios, apis, etc.

¿Qué agrupamos y que controlamos en el encapsulamiento?

Agrupamos y controlamos quienes pueden determinada información y elegimos la forma de hacerlo, podemos definir que grupos específicos acceden a algunas cosas o podemos hacer que algunas variables solo sean visibles para hijos de una clase determinada, por eso es importante encapsular.

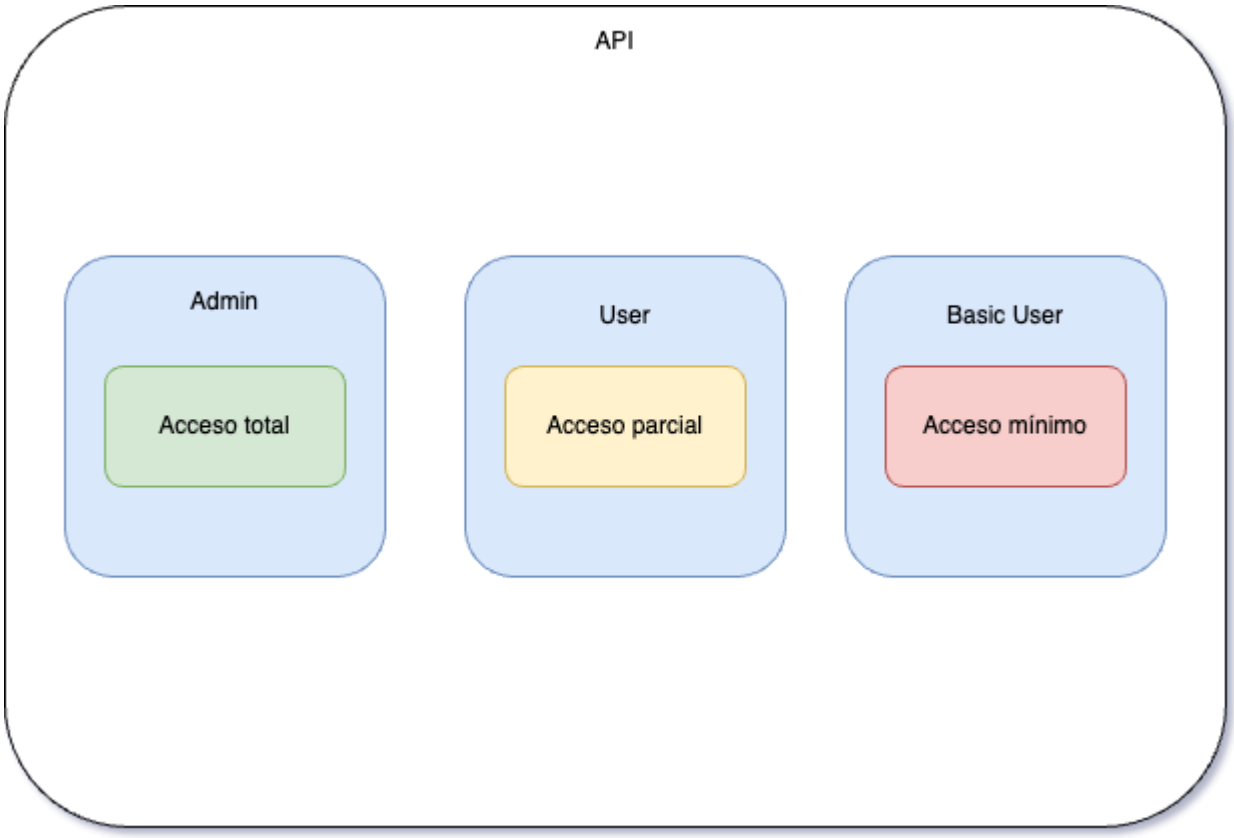
Te voy a mostrar un ejemplo con código.

```
function crearContador() {
    let contador = 0; // Variable privada

    return {
        incrementar: function() {
            contador++;
            return contador;
        },
        obtenerValor: function() {
            return contador;
        }
    };
}

const miContador = crearContador();
console.log(miContador.incrementar()); // 1
console.log(miContador.incrementar()); // 2
console.log(miContador.obtenerValor()); // 2
```

En este ejemplo de programación funcional, básicamente estamos controlando y agrupando la variable contador para que solo se pueda acceder y modificar usando las funciones incrementar y obtenerValor.

Puedes imaginar el mismo ejemplo pero en lugar de un contador, puede ser una contraseña o puede ser una tabla de base de datos, etc.

También podemos ver aplicado el encapsulamiento en los permisos de un API.



En ese ejemplo vemos como podemos encapsular la información a la que accede cada tipo de usuario un API de cualquier tipo.

También podemos ver esto mismo aplicado a nivel de SQL, y una forma muy común de hacerlo es a través de las vistas.

```
CREATE VIEW EmpleadosActivos AS
SELECT id, nombre, puesto
FROM empleados
```

```
WHERE estado = 'activo';
```

Con esa vista, podríamos dar acceso a un usuario SQL específico para que tenga acceso solo a empleados activos de una base de datos, sin exponer o arriesgar otra información.

Al final en nuestro día a día estamos rodeados de encapsulamiento de una u otra manera, y sin darnos cuenta lo aplicamos en otros estilos de programación diferentes a la POO e incluso a un nivel más de diseño como el caso de un API REST o también para definir una infraestructura, ahora imagina lo que podemos hacer si es un concepto que lo mantenemos en mente y lo aplicamos de forma más consiente.