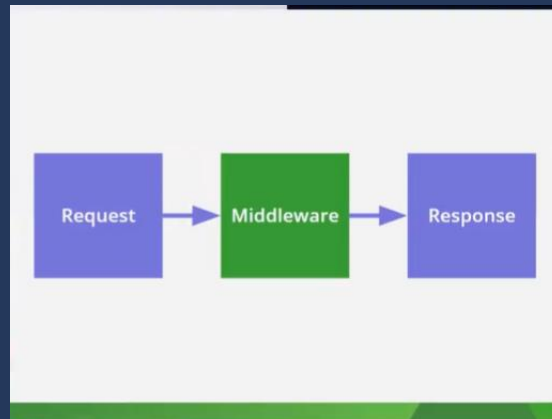


# ¿Qué son los Middlewares?

Los middlewares están en medio de los requests y los responses  
requests – middlewares – responses



Un ejemplo de middleware podría ser la función callback que recibimos de los routers de nuestra app, ya que recibe el request y el response, y podemos manejarla como lo programemos ya que está en medio de ellas.

```
app.get('/', (req, res) => console.log("Esta función es un middleware porque maneja tanto al request y al response"))
```

Los middlewares nos permiten procesar lógica, ya que cuando nos vienen los datos de un request, podemos hacer cierta lógica para presentar los datos de acuerdo a nuestra necesidad.

Podemos encadenar middlewares, es decir, que un middleware llame a otro, por ejemplo, tenemos un middleware de autenticación, que si no pasa dicha autenticación, no va a pasar al siguiente middleware, que podría ser cualquier cosa. Pero si pasa la autenticación podemos pasar al siguiente con el tercer argumento de la función middleware, el cual por convención se le llama 'next'

```
app.get('/', (req, res, next) => {  
  
  if(false) res.send('No pasaste el middleware, hasta acá llegó la solicitud')  
  
  else next()  
  
})
```

Puede también incluirse otra clase de Middlewares de tipo error, que en este caso contaría con 4 parámetros:

```
function (error, req, res, next) {  
  if (error) {  
    res.status(500).json({error});  
  } else {  
    next();  
  }  
}
```

**Casos de uso:**

- Funcionan como pipes
- Validar datos
- Capturar errores
- Validar permisos
- Controlar accesos

**Pueden seguir dos flujos dependiendo de la lógica que se implemente:**

- Request → Middleware → Response.
- Request → Middleware Y → Middleware X → Middleware Z → más Middlewares → Response