

# MIDDLEWARES A NIVEL APLICACIÓN

“

Son aquellos middlewares que queremos que se ejecuten **siempre** a lo largo de toda la aplicación, sin importar a qué ruta ingrese el usuario.



# CÓMO CONFIGURARLO

Invocando el método `app.use()` estamos configurando un middleware que se va a implementar en toda la aplicación.

Este método recibe un callback con tres parámetros:

- el objeto **request**
- el objeto **response**
- la función **next**

```
app.use(function(req, res, next) {  
    ...  
})
```

{}

Algunos de los middlewares a **nivel aplicación** que ya venimos utilizando son:

```
// configuración de carpeta de archivos estáticos
app.use(express.static(__dirname + '/public'));

// configuración de ruteo
const rutasProductos = require('./routes/products');
app.use('/', rutasProductos);
```



# QUÉ ES **NEXT**

`next` es un callback que se va a encargar de apilar todos los middlewares que apliquen a una misma petición, y **ejecutarlos** uno tras otro. Cuando llegue al último y, si se ejecutaron correctamente, pasará al siguiente paso que es llegar al controlador que maneja esa ruta.

*Por eso siempre al terminar cada middleware, ejecutamos **next**.*

```
app.use(function(req, res, next) {  
    ...  
    next();  
})
```



Configuremos un middleware que maneje los errores **404** de la aplicación.



```
{ código }
```

```
app.use((req, res, next) => {  
  res.status(404).render('404-page');  
  next();  
});
```

```
{ código }
```

```
app.use();
```

En el **entry point** de la aplicación, llamo al método **use()** para configurar un middleware que aplique a todas las peticiones del sitio.



```
{ código }
```

```
app.use((req, res, next) => {  
    ...  
});
```

Le paso el callback con los tres parámetros necesarios: el request, el response y el **next**.

```
{ código }
```

```
app.use((req, res, next) => {  
  res.status(404).render('404-page');  
});
```

Si la página **no existe**, devolverá un error de status 404 y, adicionalmente, renderizará la vista que tenga diseñada para ese escenario.

```
{ código }
```

```
app.use((req, res, next) => {  
    res.status(404).render('404-page');  
    next();  
});
```

La función **next** se encargará de ejecutar el próximo paso. Si la página **existe**, llamará al controlador y este devolverá la vista solicitada.