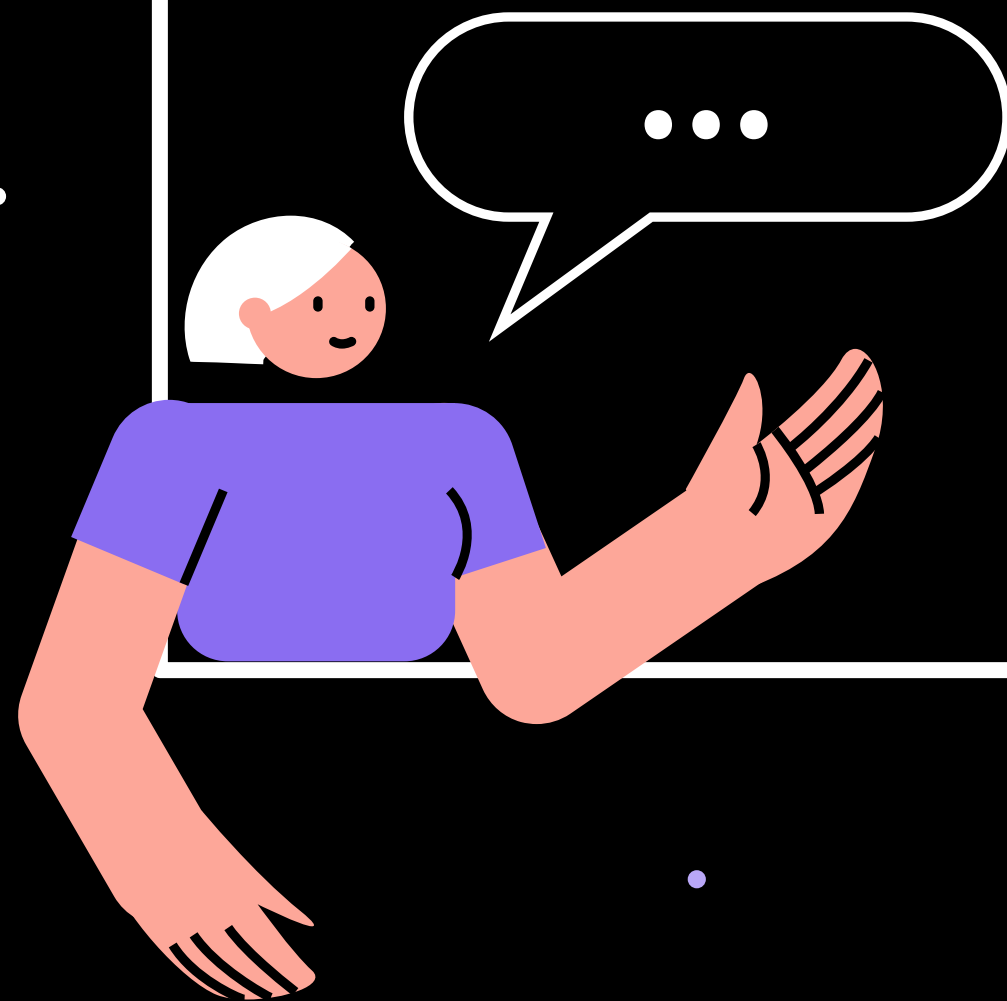


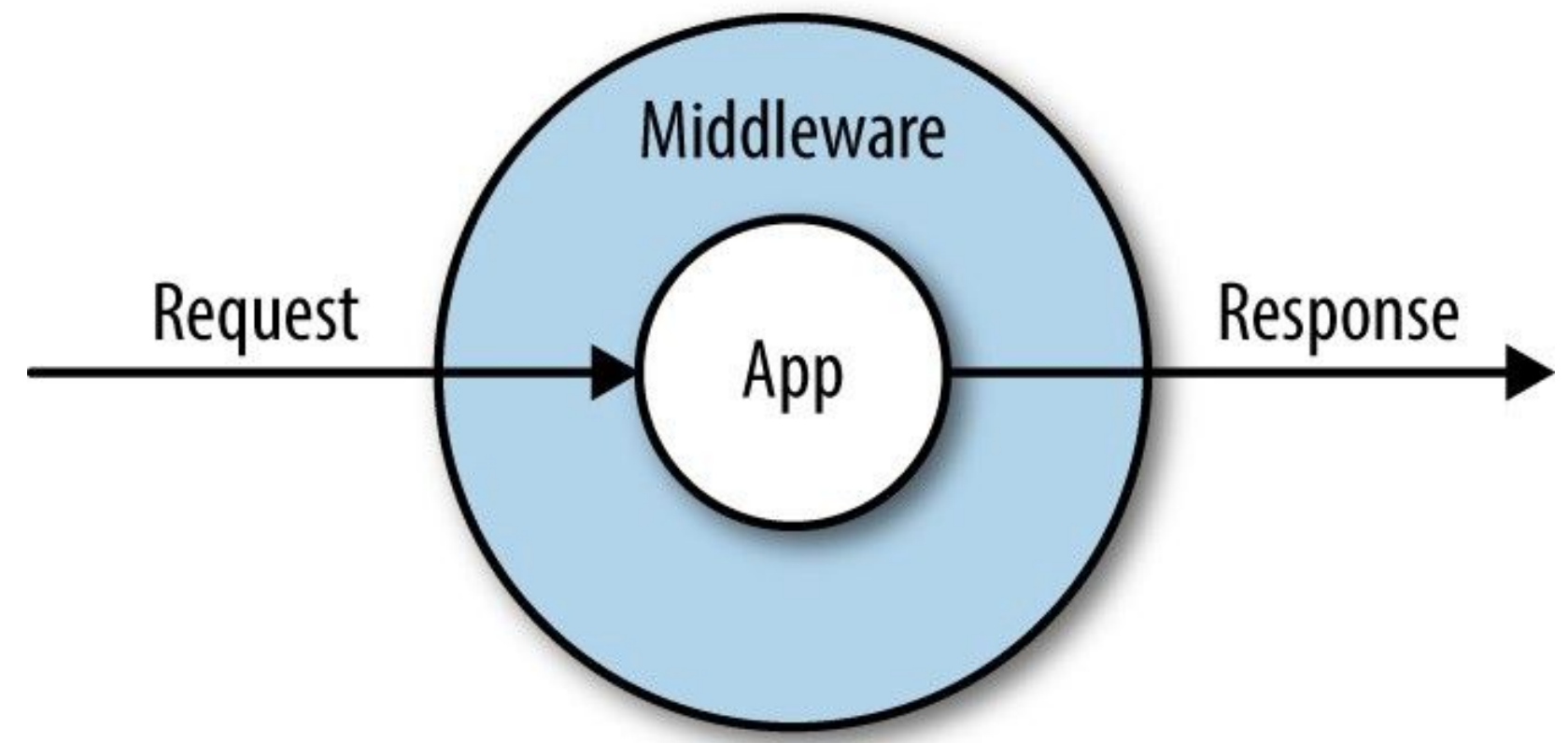
# Autenticación en Go



En términos de arquitectura del software, un middleware es un componente en nuestras aplicaciones que nos va a permitir **centrarnos en la lógica de negocio**, resolviendo algunas complejidades de bajo nivel.

# Middlewares en un contexto HTTP

Las funciones middleware en un contexto HTTP pueden tener propósitos muy amplios y diferentes.





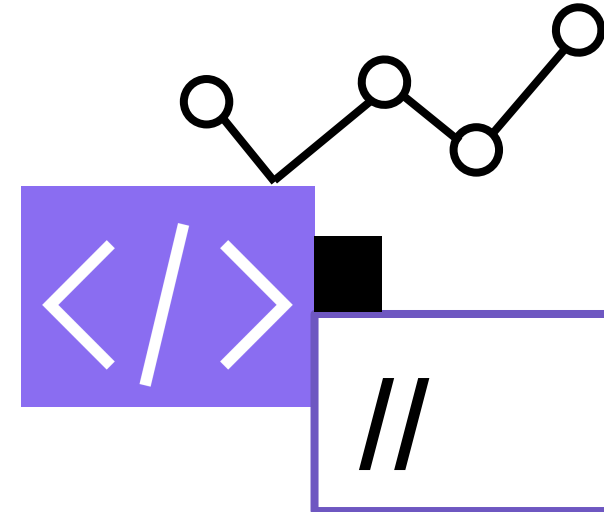
# Middlewares en Go

```
{}  
  
func GetDummyEndpoint(c *gin.Context) {  
    resp := map[string]string{"hello": "world"}  
    c.JSON(200, resp)  
}  
  
func main() {  
    api := gin.Default()  
    api.GET("/dummy", GetDummyEndpoint)  
    api.Run(":8080")  
}
```



# Middlewares en Go

```
{}  
  
func DummyMiddleware(c *gin.Context) {  
    fmt.Println("Im a dummy!")  
  
    // Pass on to the next-in-chain  
    c.Next()  
}  
  
func main() {  
    // Insert this middleware definition before any routes  
    api.Use(DummyMiddleware)  
    // ... more code  
}
```



# Middlewares en Go

```
{}  
  
func DummyMiddleware() gin.HandlerFunc {  
    // Do some initialization logic here  
    // Foo()  
    return func(c *gin.Context) {  
        c.Next()  
    }  
}  
  
func main() {  
    // ...  
    api.Use(DummyMiddleware())  
    // ...  
}
```

# Middleware de autenticación de API

Si estamos creando una **API con Gin**, probablemente queramos agregar algún tipo de mecanismo de autenticación. La solución más sencilla es comprobar si el cliente nos envía un parámetro de URL, como **api\_token**, para luego validarlo y dejarlo realizar el request.





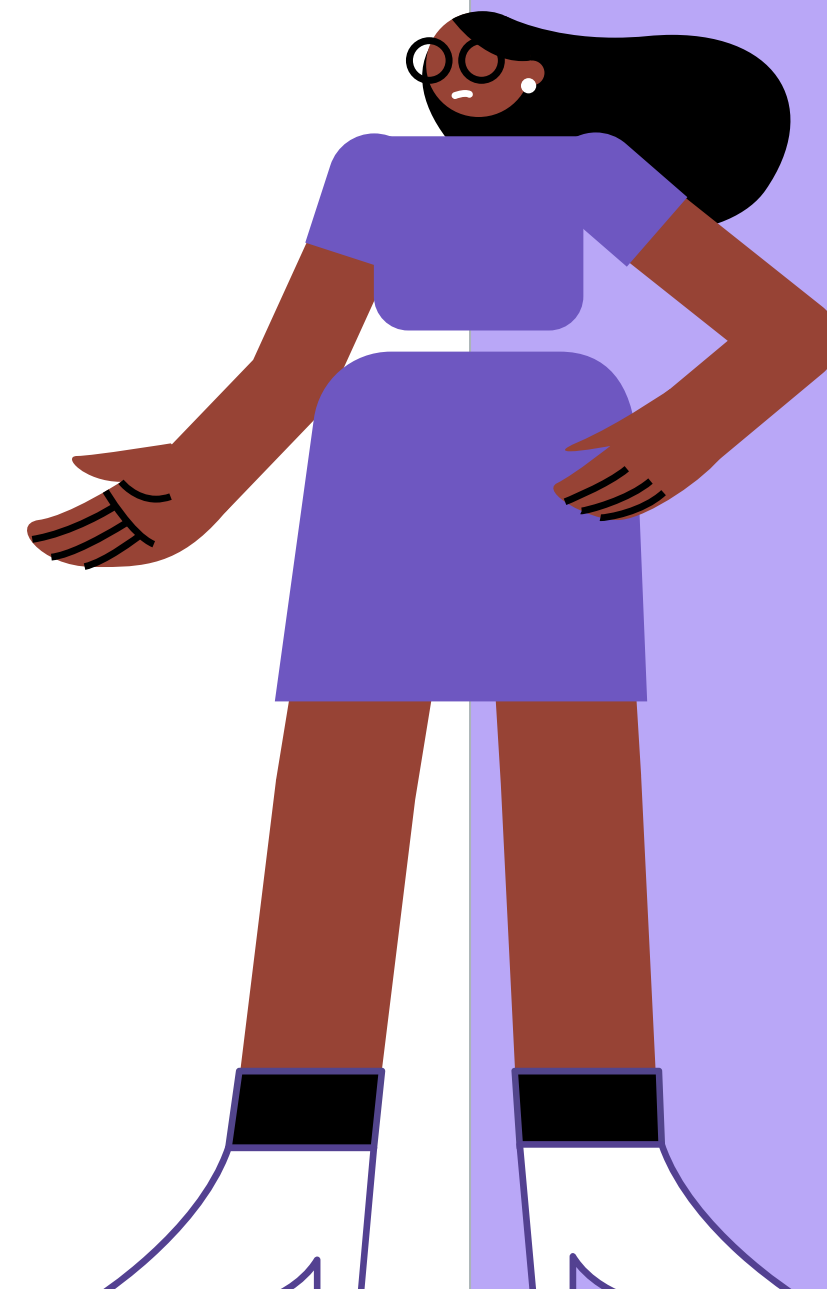
# Middleware de autenticación de API

```
func TokenAuthMiddleware() gin.HandlerFunc {  
    requiredToken := os.Getenv("API_TOKEN")  
  
    return func(c *gin.Context) {  
        token := c.GetHeader("api_token")  
  
        if token != requiredToken {  
            c.AbortWithStatusJSON(401, gin.H{"error": "Invalid API token"})  
            return  
        }  
  
        c.Next()  
    }  
}
```



# Conclusiones

Aprendimos a implementar middlewares para poner capas intermedias en nuestras aplicaciones, como un sistema de validación.



¡Muchas gracias!