

東京' TOKIOTA

Aspnet core / Middlewares

Sesión 2 - Acciona





Jose Manuel Corral

Cloud & Development Lead Consultant

jose.corral@Tokiota.com

 • @jmanuelcorralcorral

ASP.NET Core

- ~~Monolitico~~
- Funciona y podemos desarrollar en diferentes plataformas.
- Soporte para Roslyn
- Servidor web Kestrel
- CLI



Conceptos Básicos

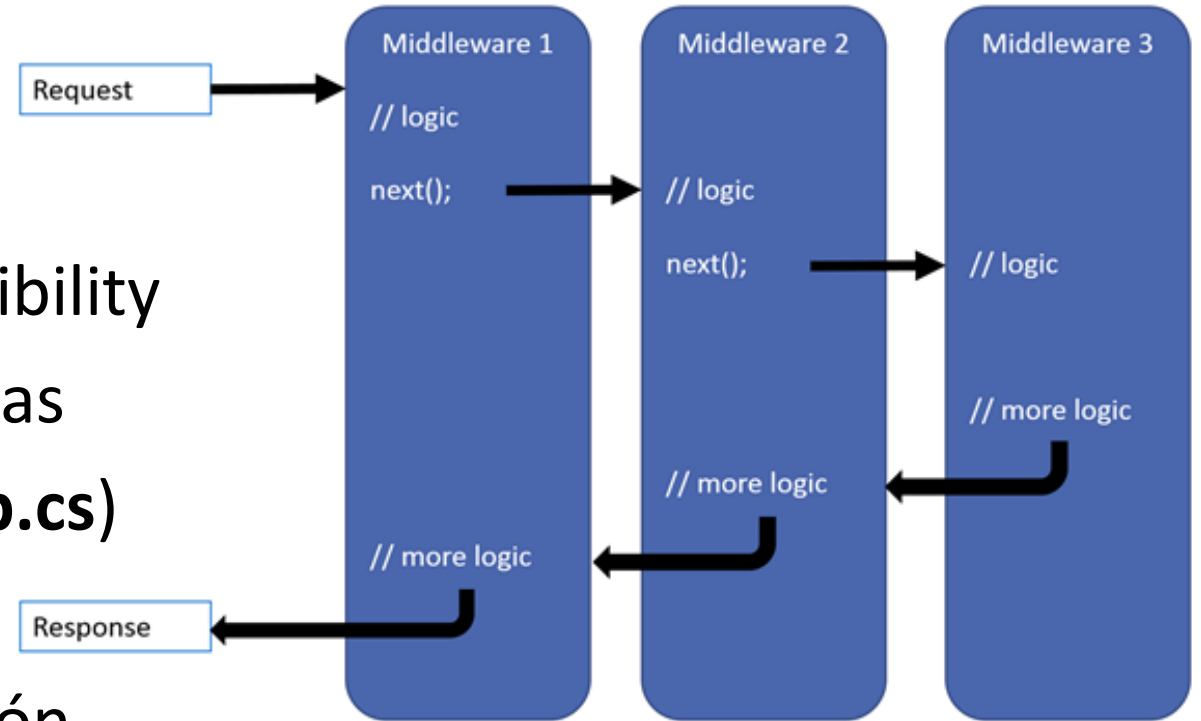
- DI
- IoC
- Service Locator antipattern

Nuevos Conceptos

- Startup class
- IApplicationBuilder Interface
- IHostingEnvironment Interface
- Project.json

Middlewares

- Siguen el Patrón: Chain of Responsibility
- Soportan Inyección de Dependencias
- Se invocan en el Configure (**Startup.cs**)
- Invocados cada petición
- Pueden “Cortocircuitar” una petición
- Son capaces de crear la propia HTTP Response



Middleware Interesantes

- Microsoft.aspnet.diagnostics
- Microsoft.aspnet.staticfiles
- Microsoft.aspnet.mvc.core
- Microsoft.aspnet.mvc
- Microsoft.AspNetCore.All (polémico)

Posibles Implementaciones Middlewares

- Tracking de las acciones que hace un usuario.
- Handler de excepciones.
- Autenticación y Autorización.
- Caché

Ejercicio



Ejercicio

```
$ cd multiplicaciones
```

```
$ dotnet new web -n "multi.web" -o "src/multi.web"
```

```
$ dotnet sln multi.sln add "src/multi.web/multi.web.csproj"
```

```
$ code .
```

Ejercicio

- Refactorizaremos la funcionalidad realizada el otro día, para que esté en una librería, con esto será posible invocarla tanto desde la web como desde consola.
- Arreglamos los Tests que ahora fallan.
- Invocaremos desde el middleware.
- Comprobaremos que devuelve.
- Test Integración.

Middleware

```
1 namespace multi.web
2 {
3     using System;
4     using System.Threading.Tasks;
5     using Microsoft.AspNetCore.Http;
6     using Microsoft.Extensions.Options;
7     using multi.utils;
8     0 references
9     public class SampleMiddleware
10    {
11        2 references
12        private readonly RequestDelegate next;
13
14        0 references
15        public SampleMiddleware(RequestDelegate next)
16        {
17            this.next = next ?? throw new ArgumentNullException(nameof(next));
18        }
19
20        0 references
21        public async Task Invoke(HttpContext context)
22        {
23            //Hacemos la nuestra
24            await next(context);
25        }
26    }
27 }
```

Añadiendo los tests

```
$ cd multiplicaciones
```

```
$ dotnet new xunit -n "multi.web.tests" -o "tests/multi.web.tests"
```

```
$ dotnet add "tests/multi.web.tests/multi.web.tests.csproj" reference  
"src/multi.web/multi.web.csproj"
```

```
$ dotnet sln multi.sln add "tests/multi.web.tests/multi.web.tests.csproj"
```

```
$ code .
```

Instalando packages de nuget

```
$ cd multiplicaciones
```

```
$ cd tests/multi.web.tests
```

```
$ dotnet add package Microsoft.AspNetCore.TestHost
```

```
$ code .
```

東京' TOKIOTA