

Sesión 01

Introducción a Minimal API en Net 8

Instructor:

ERICK ARÓSTEGUI

earostegui@galaxy.edu.pe



8
NET
FULL-STACK
DEVELOPER

ÍNDICE

- 01** ASP.NET Core novedades y mejoras

- 02** Minimal APIs vs Controller APIs(Classic)

- 03** Arquitectura de Minimal APIs

- 04** Application/WebApplicationBuilder y Routing

- 05** Integración de Routes, Verbs(MapPost, MapPut y MapDelete) y HTTP Status Codes

- 06** Desarrollando y desplegando pruebas de concepto(PoC)

01

ASP.NET Core novedades y mejoras



Arquitectura de ASP.NET Core

Arquitectura de ASP.NET Core

ASP.NET Core Platforms

Kestrel

Model binding

Middleware

Dependency Injection

Razor

Logging

Arquitectura de ASP.NET Core

ASP.NET Core Application Platforms

ASP.NET Core
MVC

ASP.NET Core
Razor Pages

ASP.NET Core
API &
Minimal APIs

ASP.NET Core
Blazor

ASP.NET Core Platforms

Kestrel

Model binding

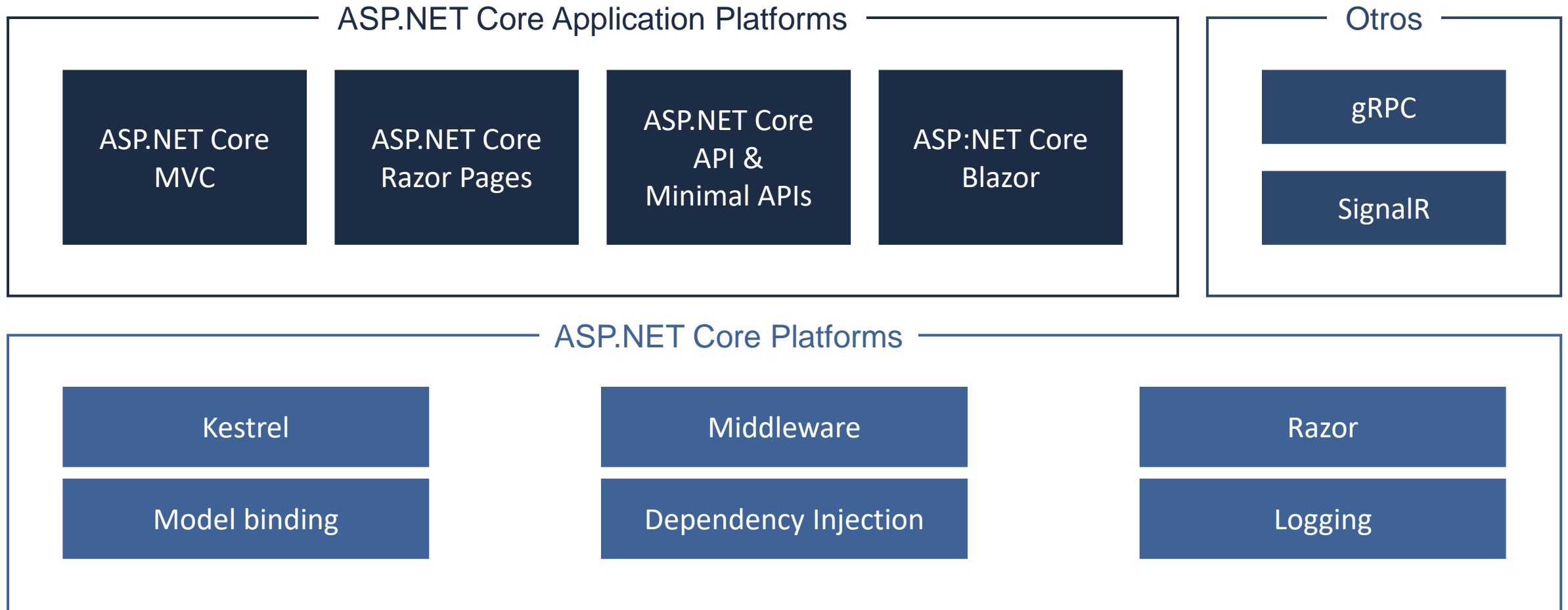
Middleware

Dependency Injection

Razor

Logging

Arquitectura de ASP.NET Core



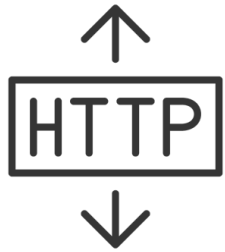
Mejoras principales en ASP.NET Core 7

Mejoras de
rendimiento

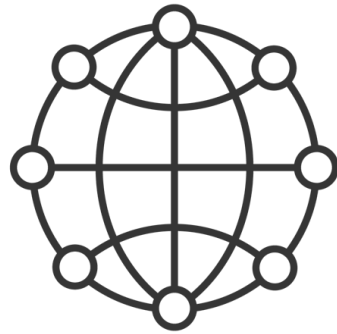
Adiciones a Blazor y
Minimal APIs

Cambios en el
servidor y en el
runtime

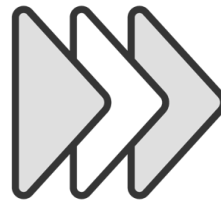
Nuevas características de ASP.NET Core 7



HTTP/2
WebSockets



HTTP/3



Output caching

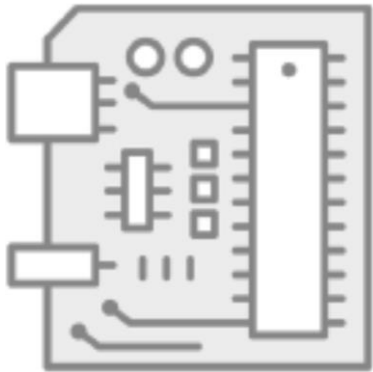


Rate limiting



Request
decompression

Nuevas características de ASP.NET Core 8



Route
short-circuiting



Keyed service
registration



Soporte nuevas
métricas



Mejora de los
background
services



.http files en
Visual Studio

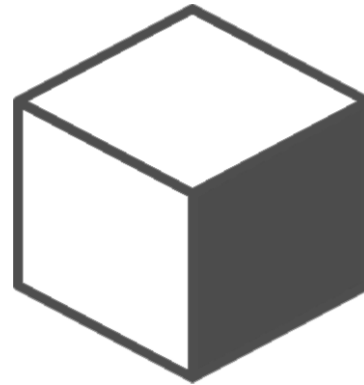
Nuevas características de ASP.NET Core 8



Cambios en
Identity



HTTP/3 por
defecto



Native AOT

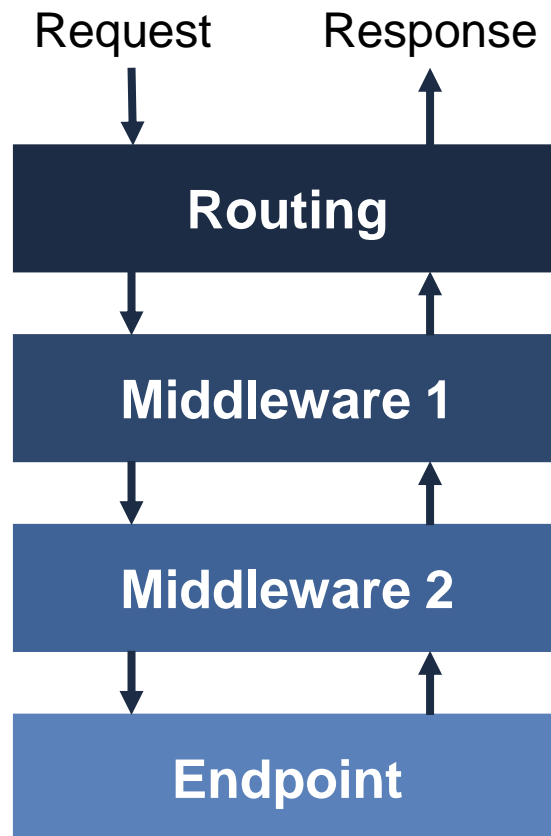


Form binding
en Minimal
APIs

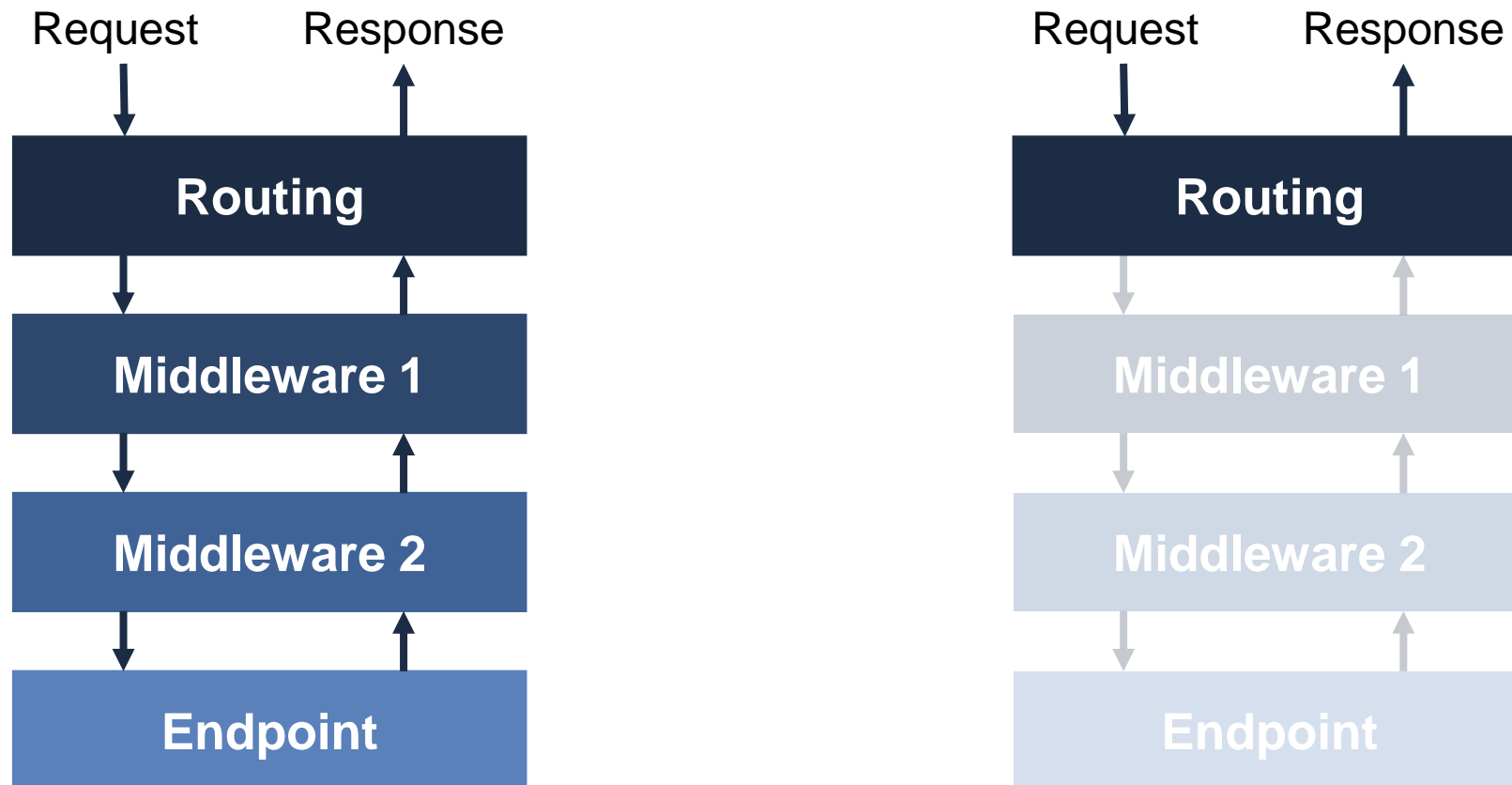


Depuración
mejorada

Route Short-circuit



Route Short-circuit



NET Compilation Workflows

Standard .NET compilation & execution

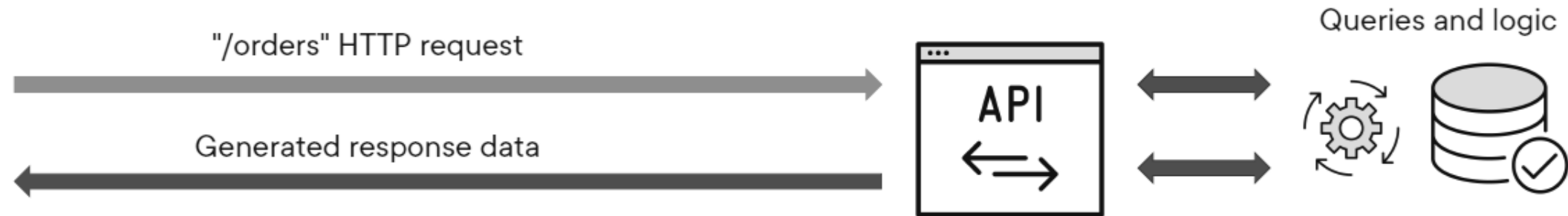


AOT compilation & execution



Output Caching

First request



Second request



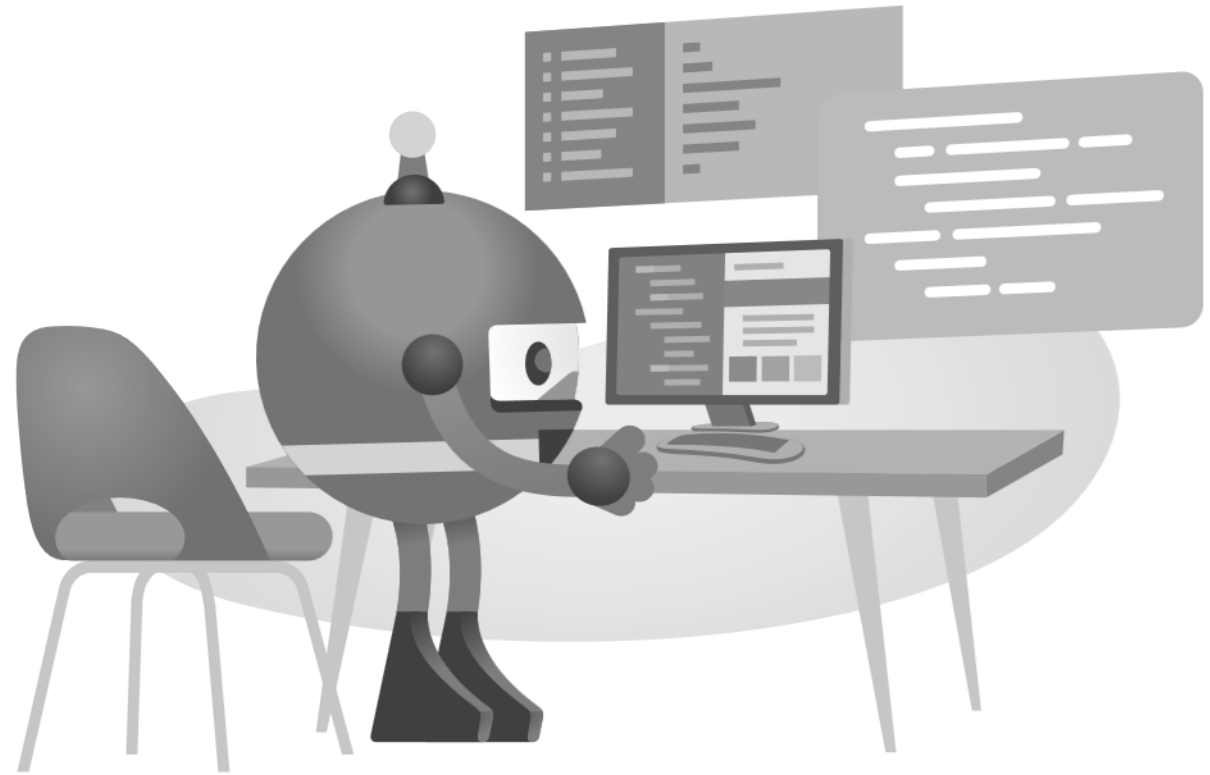
Output Caching

```
builder.Services.AddScoped<IMenuService, MenuService>();  
builder.Services.AddOutputCache();  
builder.Services.AddStackExchangeRedisOutputCache(options =>  
{  
    options.Configuration = "your_connection_string";  
    options.InstanceName = "dotnet8redis";  
});  
builder.Services.AddMvc().AddNewtonsoftJson();
```

```
app.MapGet("/menu", (IMenuService menuService) =>  
{  
    return menuService.GetMenuItems();  
})  
    .CacheOutput();
```


ASP.NET Core novedades y mejoras

DEMO



02

Minimal APIs vs Controller APIs(Classic)



→ Minimal APIs vs Controller APIs(Classic)

```
var app = WebApplication.Create(args);  
app.MapGet("/person", () => new Person("Gill", "Cleeren")); await  
app.RunAsync();
```

```
public record Person(string FirstName, string LastName);
```

Un muy Minimal API

→ Minimal APIs vs Controller APIs(Classic)

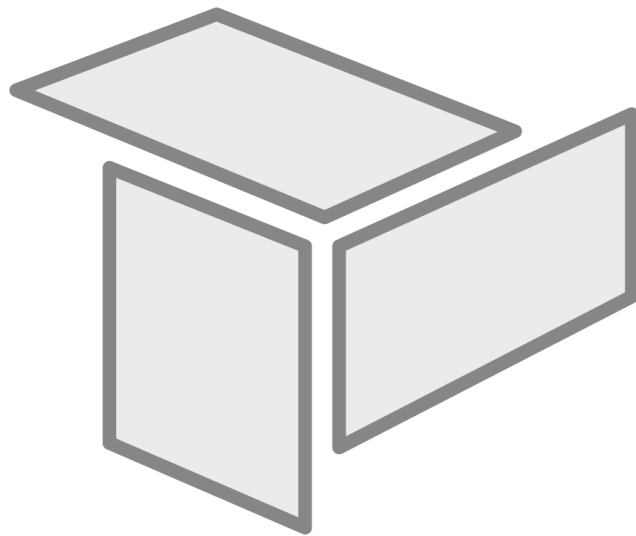
Plataforma madura

Endpoint filters

Route groups

TypedResults

Endpoint filters



- Ejecutar código antes y después del controlador
- Puede inspeccionar y cambiar parámetros.
- Útil para preocupaciones transversales
 - Inicio sesión
 - Validación

Endpoint filters

```
app.MapGet("/todos", (IToDoItemRepository toDoItemRepository) =>
{
    return toDoItemRepository.GetAllToDoItems();
}).AddEndpointFilter(async (context, next) =>
{
    //do something here like logging return await next(context);
    return await next(context);
});
```

Route groups

```
app.MapGroup("/api").AddEndpointFilter(async (context, next) =>
{
    ...
});
```

TypedResults

```
app.MapGet("/todos/{id}", async (int id, TodoDb db) =>  
    await db.Todos.FindAsync(id)  
    is Todo todo  
    ? TypedResults.Ok(todo)  
    : TypedResults.NotFound());
```


03



Arquitectura de Minimal APIs

Opciones para estructurar Minimal APIs

Opciones para estructurar Minimal APIs



Usar métodos en lugar de
controladores en línea

Opciones para estructurar Minimal APIs



Usar métodos en lugar de controladores en línea



Separación de métodos de controlador en clases

Opciones para estructurar Minimal APIs



Usar métodos en lugar de controladores en línea



Separación de métodos de controlador en clases



Extensión
IEndpointRouteBuilder

Opciones para estructurar Minimal APIs



Usar métodos en lugar de controladores en línea



Separación de métodos de controlador en clases



Extensión
IEndpointRouteBuilder



Combinando 1, 2 y 3

Opciones para estructurar Minimal APIs



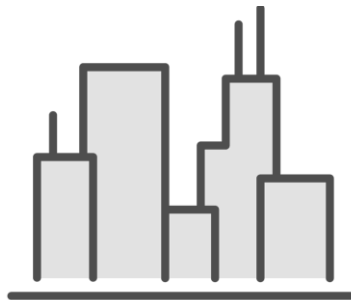
Usar métodos en lugar de controladores en línea



Separación de métodos de controlador en clases



Extensión
IEndpointRouteBuilder



Combinando 1, 2 y 3



Librería de terceros

Opciones para estructurar Minimal APIs



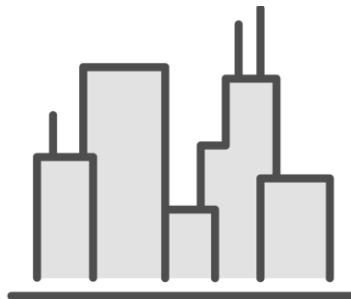
Usar métodos en lugar de controladores en línea



Separación de métodos de controlador en clases



Extensión
IEndpointRouteBuilder



Combinando 1, 2 y 3



Librería de terceros



Otros enfoques

04

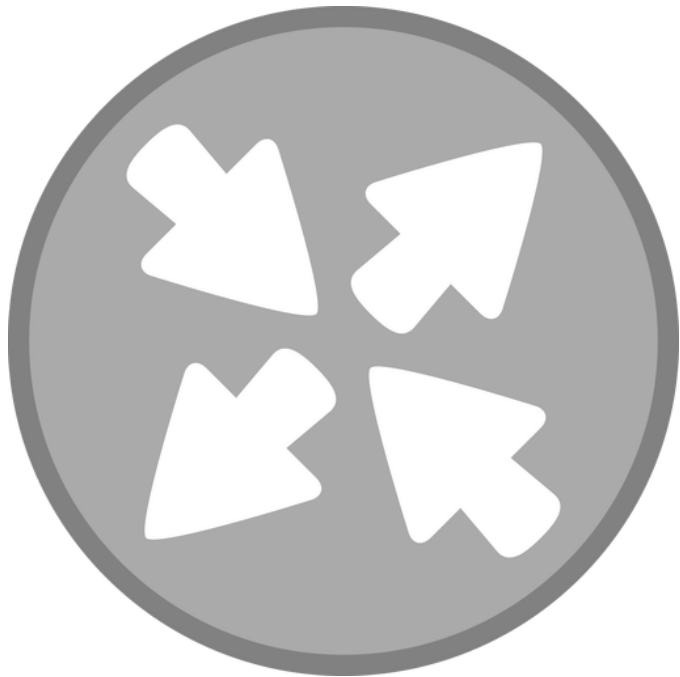
Application/WebApplicationBuilder y Routing



Routing

El proceso de hacer coincidir un método HTTP y URI con un controlador de ruta específico

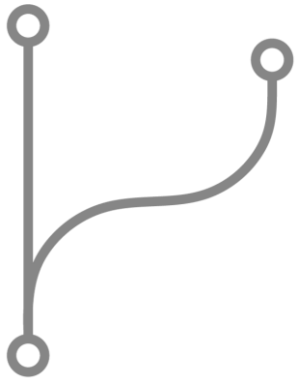
Routing



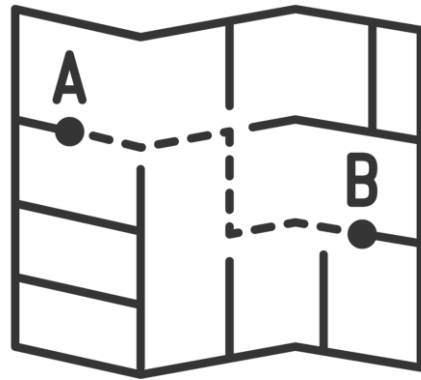
app.**MapAction** methods

- Donde **Action** = El metodo HTTP
- IEndpointRouteBuilder

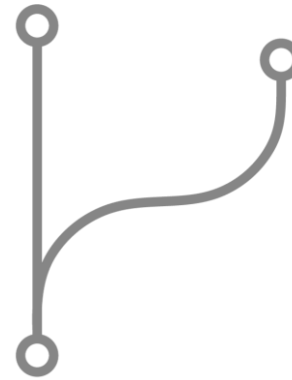
Conceptos de Routing



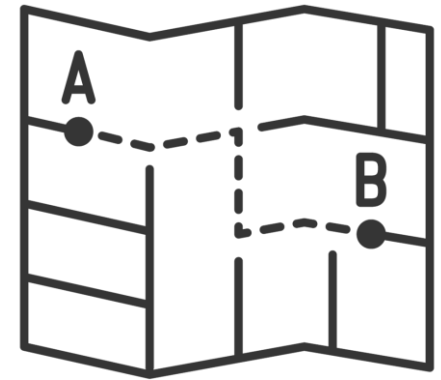
Metodo Route builder
de **IEndpointRoute
Builder**



Plantillas o patrones
de Rutas



Parámetros de Rutas



Restricciones de
Rutas

Parametros Binding Sources

Routes Values
[FromRoute]

Query string
[FromQuery]

Header
[FromHeader]

Body (as JSON)
[FromBody]

**Services provider
By DI**
[FromServices]

Custom

05

Integración de Routes, Verbs(MapPost, MapPut y MapDelete) y HTTP Status Codes

Routing

Método HTTP	Endpoint route builder method	Request payload	Ejemplo de URI	Response payload
GET	MapGet	-	/authors /authors/{authorId}	Colección de authors Un único author
POST	MapPost	Un único author	/authors	Un único author
PUT	MapPut	Un único author	/authors/{authorId}	Un único author o vacío
DELETE	MapDelete	-	/authors/{authorId}	-

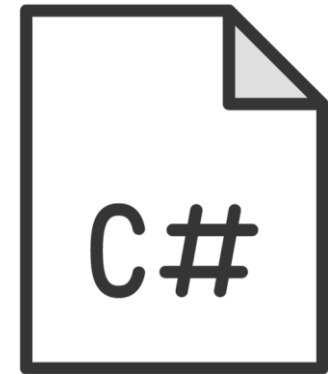
Valores retornables de un Minimal API

[A, B, C]

string



Cualquier tipo, pero
se genera una cadena



IResult-based types:
Results.X,
TypedResults.X

Codigo de Estados HTTP mas comunes

200 – Ok
201 – Created
204 – No Content

400 – Bad Request
401 – Unauthorized
403 – Forbidden
404 – Not Found
405 – Method Not
Allowed

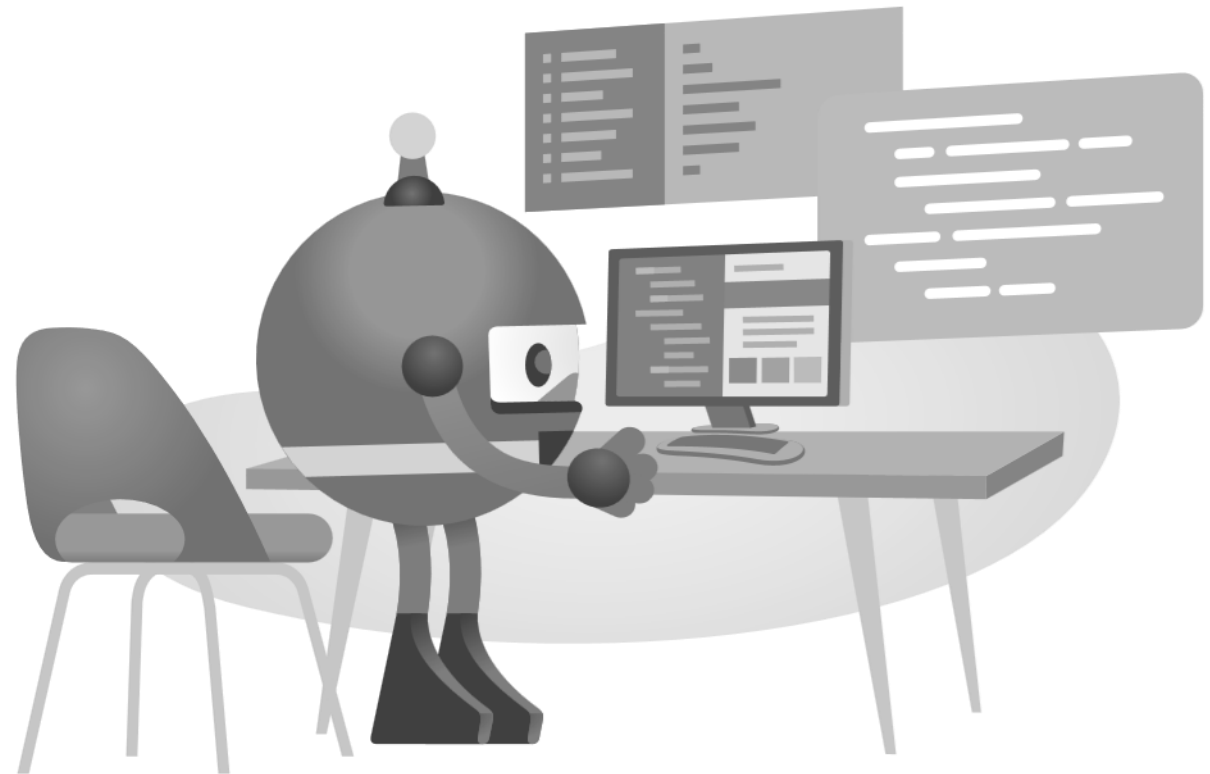
500 – Internal Server
Error

06

Desarrollando y desplegando pruebas de concepto(PoC)

Desarrollando y desplegando pruebas de concepto(PoC)

DEMO



A man with a beard, wearing a striped shirt, is sitting at a desk. He is looking at a laptop screen and has a pen in his mouth. The background is dark blue with a large, light blue circular shape behind the man.

GRACIAS

POR SU PREFERENCIA

