Sesión 01

Introducción a Minimal API en Net 9

Instructor:

ERICK ARÓSTEGUI

earostegui@galaxy.edu.pe

GALAXY
TRAINING

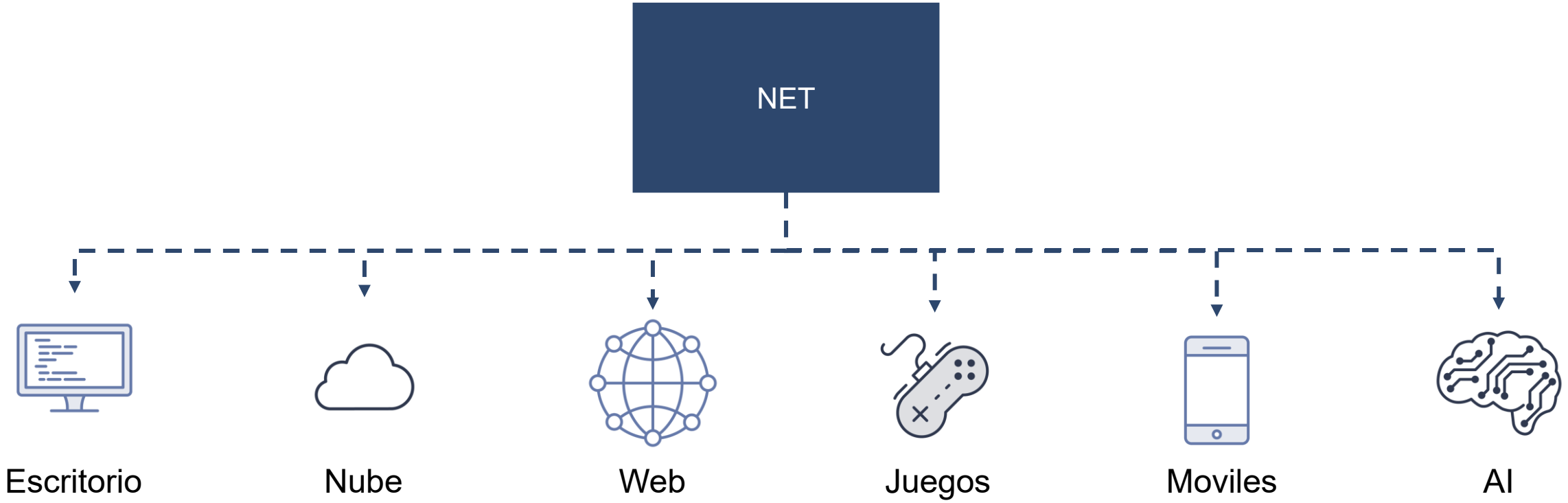
ÍNDICE

GALAXY
TRAINING

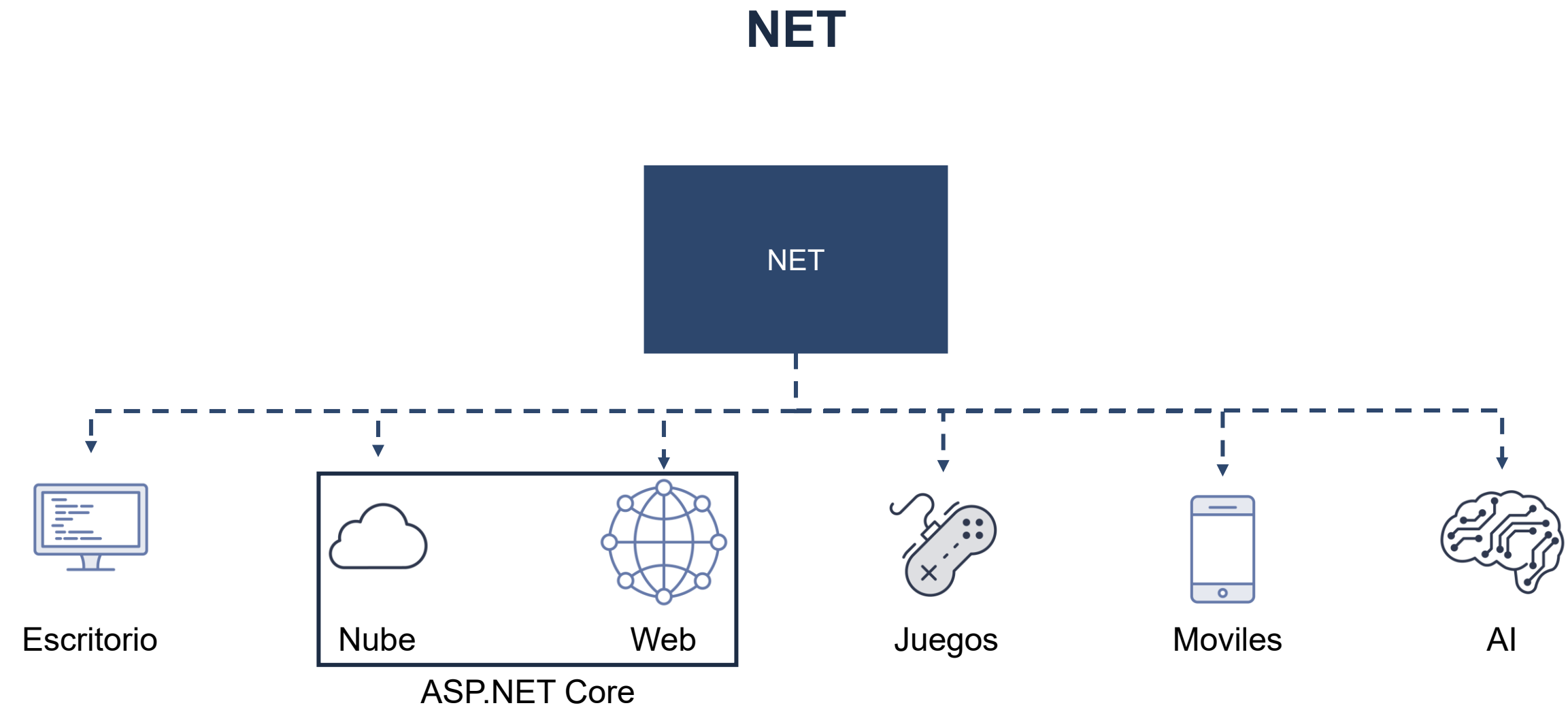
01

ASP.NET Core novedades y mejoras

NET



ASP.NET Core novedades y mejoras

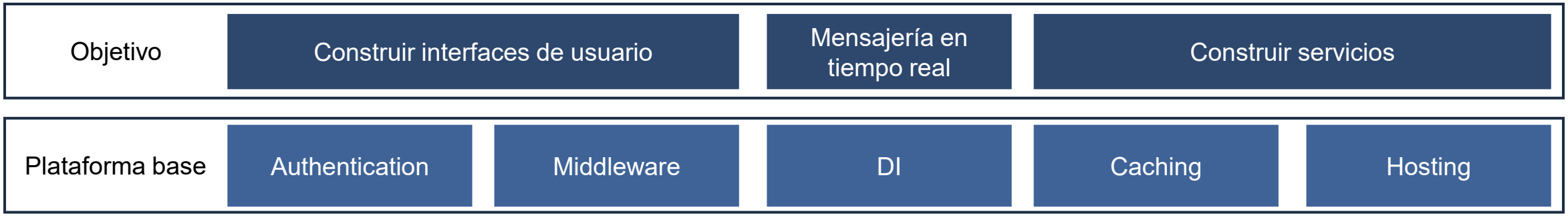


Patrones de Aplicación en ASP.NET Core

GALAXY TRAINING

Patrones de Aplicación en ASP.NET Core

Patrones de Aplicación en ASP.NET Core



GALAXY TRAINING

Patrones de Aplicación en ASP.NET Core

Uso	Aplicaciones del lado del servidor con interfaces de usuario basadas en páginas	Interfaces de usuario ricas del cliente y del servidor	Comunicación Web socket	Servicios web HTTP y APIs	Framework para llamadas a procedimientos remotos (RPC)
Objetivo	Construir interfaces de usuario		Mensajería en tiempo real	Construir servicios	
Plataforma base	Authentication	Middleware	DI	Caching	Hosting

GALAXY TRAINING

Patrones de Aplicación en ASP.NET Core

Tecnología	MVC & Razor Pages	Blazor	SignalR	Minimal & Web APIs	gRPC
Uso	Aplicaciones del lado del servidor con interfaces de usuario basadas en páginas	Interfaces de usuario ricas del cliente y del servidor	Comunicación Web socket	Servicios web HTTP y APIs	Framework para llamadas a procedimientos remotos (RPC)
Objetivo	Construir interfaces de usuario		Mensajería en tiempo real	Construir servicios	
Plataforma base	Authentication	Middleware	DI	Caching	Hosting

GALAXY TRAINING

Mejoras principales en ASP.NET Core 8

Mejoras de rendimiento

Adiciones a Blazor y Minimal APIs

Cambios en el servidor y en el runtime

GALAXY TRAINING

Mejoras principales en ASP.NET Core 9

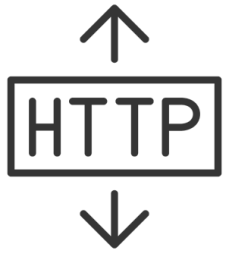
Mejoras en utilidades de Blazor y experiencia del desarrollador

Soporte nativo de OpenAPI para Minimal APIs y Controller APIs

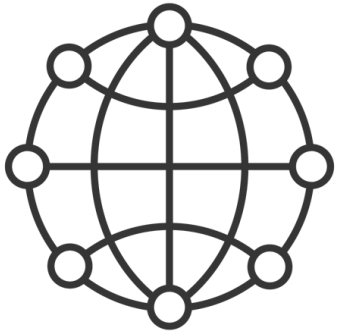
Mejoras de la plataforma para caching y archivos estáticos

Mejoras diversas en depuración, AoT, herramientas y más

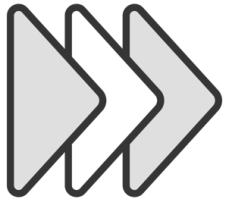
Nuevas características de ASP.NET Core 7



HTTP/2
WebSockets



HTTP/3

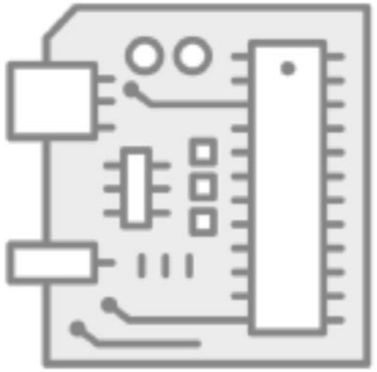


Output caching

Rate limiting

Request
decompression

Nuevas características de ASP.NET Core 8



Route short-circuiting

Keyed service registration

Soporte nuevas métricas

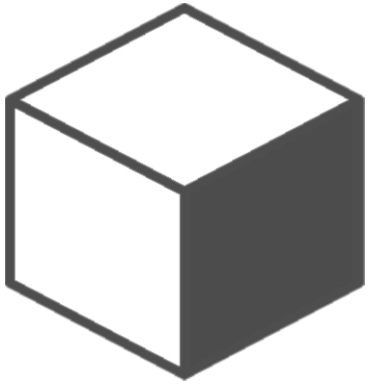
Mejora de los background services

.http files en Visual Studio

Nuevas características de ASP.NET Core 8

Cambios en Identity

HTTP/3 por defecto

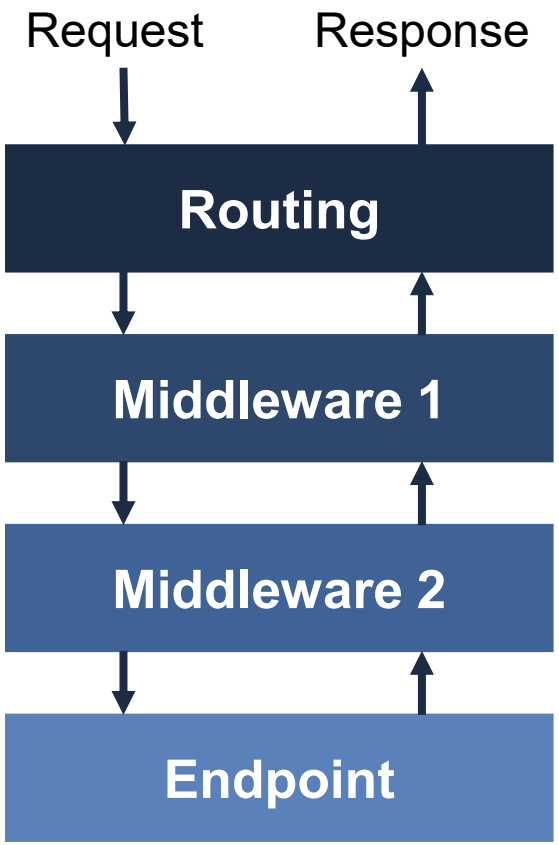


Native AOT

Form binding en Minimal APIs

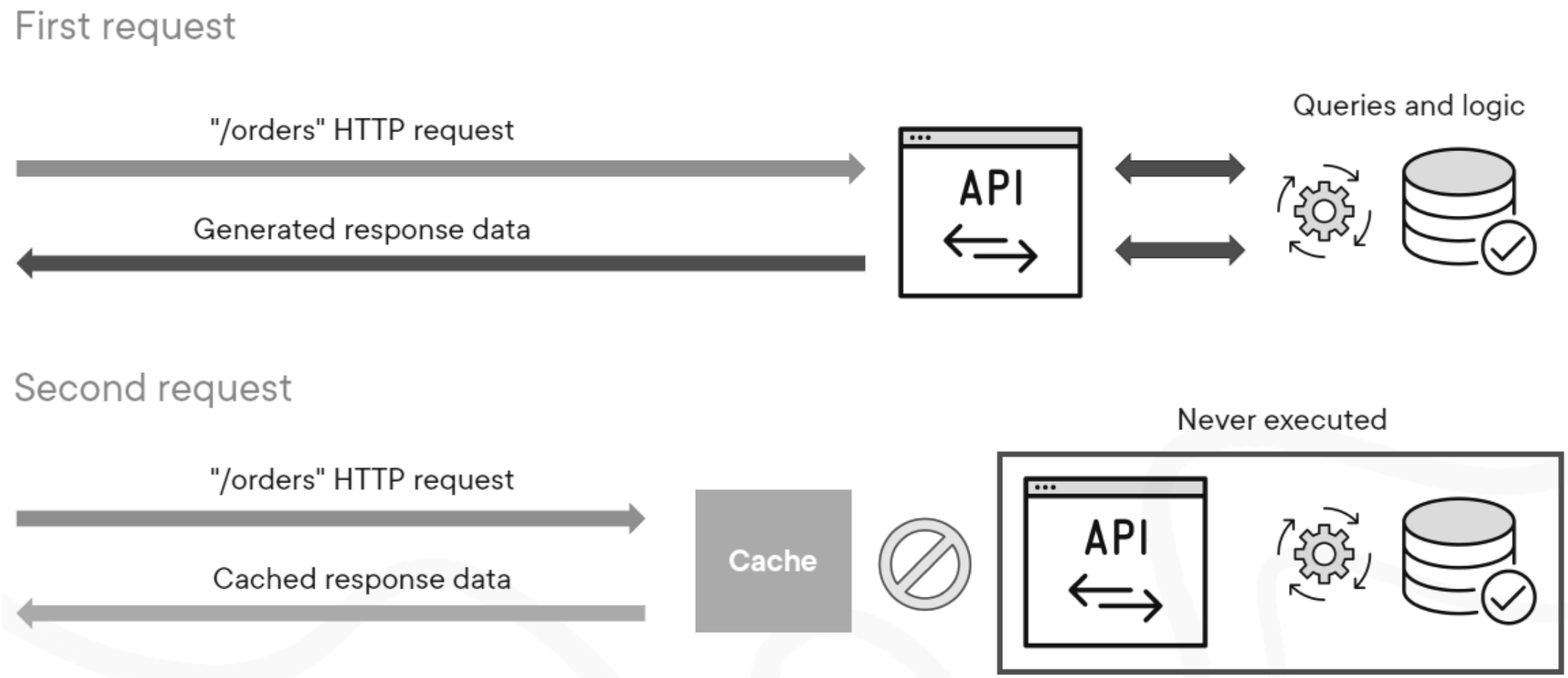
Depuración mejorada

Route Short-circuit - ASP.NET Core 8



Route Short-circuit - ASP.NET Core 8

Output Caching - ASP.NET Core 8



GALAXY TRAINING

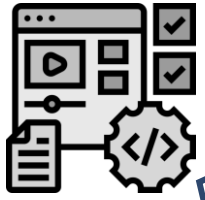
Output Caching - ASP.NET Core 8

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

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

Nuevas características de ASP.NET Core 9

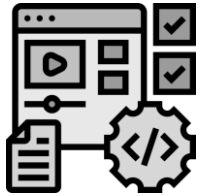
Nuevas características de ASP.NET Core 9



Mejoras significativas en la forma en que ASP.NET Core gestiona y sirve archivos estáticos como JavaScript, CSS e imágenes. Se optimiza el almacenamiento en caché, la compresión y la entrega en escenarios de alta demanda.

- **Caché eficiente**: Los archivos estáticos se sirven más rápido a través de políticas avanzadas de almacenamiento en caché.
- **Compresión optimizada**: Reducción del tamaño de archivos para acelerar el tiempo de carga.
- **Escalabilidad**: Maneja mejor grandes volúmenes de tráfico, ideal para aplicaciones con millones de usuarios.

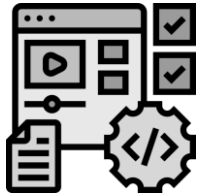
Nuevas características de ASP.NET Core 9



Blazor incorpora optimizaciones en WebAssembly y Server, con nuevas herramientas de desarrollo y mejoras en la carga de recursos. Ahora permite integrar bibliotecas de terceros más fácilmente y mejora el rendimiento en tiempo de ejecución.

- **Rendimiento mejorado:** Las aplicaciones de Blazor WebAssembly cargan más rápido gracias a una menor huella inicial.
- **Integración simplificada**: Compatible con bibliotecas modernas y herramientas de frontend.
- **Soporte extendido**: Mejor manejo de aplicaciones PWA (Progressive Web Apps).

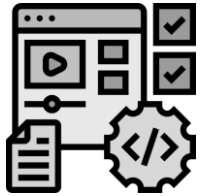
Nuevas características de ASP.NET Core 9



Mejoras en la confiabilidad y escalabilidad de SignalR, con soporte ampliado para WebSockets y mejoras en la latencia de conexión. SignalR ahora integra diagnósticos avanzados para soluciones en tiempo real.

- **Latencia reducida**: Respuestas más rápidas en escenarios de tiempo real como chats y notificaciones.
- **Escalabilidad mejorada**: Capacidad de manejar miles de conexiones simultáneas en entornos distribuidos.
- **Monitoreo avanzado**: Mejor integración con herramientas de diagnóstico como Application Insights.

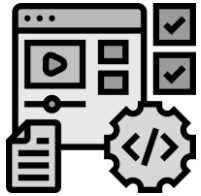
Nuevas características de ASP.NET Core 9



Minimal APIs permite crear endpoints RESTful con menos configuración y menos código. Se optimiza para casos de uso simples y microservicios. Ahora incluye soporte nativo para validación de datos y características avanzadas de enrutamiento

- **Configuración mínima**: Reducción significativa del tiempo necesario para implementar APIs.
- **Validación incorporada**: Menor necesidad de herramientas externas para validación de datos.
- **Rutas más flexibles**: Soporte avanzado para controladores y rutas dinámicas.

Nuevas características de ASP.NET Core 9



Optimización de Entrega de Recursos

Mejoras en Blazor

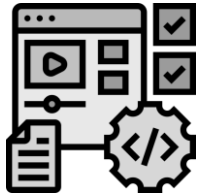
Mejoras en SignalR

APIs Ligeras con Minimal APIs

Soporte nativo para la generación de documentación con OpenAPI, compatible con Minimal APIs y APIs basadas en controladores. Facilita la interoperabilidad al exponer las especificaciones de tus APIs automáticamente.

- **Generación automática**: Produce especificaciones JSON compatibles con Swagger UI sin configuración manual.
- **Interoperabilidad**: Mejora la compatibilidad con clientes API en diferentes plataformas.
- **Facilidad de uso**: Reduce el tiempo invertido en documentar servicios.

Nuevas características de ASP.NET Core 9



Optimización de Entrega de Recursos

Mejoras en Blazor

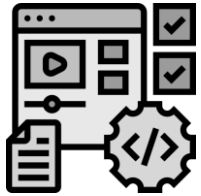
Mejoras en SignalR

APIs Ligeras con Minimal APIs

Mejoras en el manejo de flujos de autenticación con soporte ampliado para OpenID Connect, OAuth2 y autenticación por tokens. Permite políticas de autorización más específicas y flexibles.

- **Seguridad avanzada**: Mejor protección contra accesos no autorizados.
- **Integración sencilla**: Compatible con proveedores modernos como Azure AD y Google Identity.
- **Autorización granular**: Define políticas específicas para cada ruta o recurso.

Nuevas características de ASP.NET Core 9



Optimización de Entrega de Recursos

Mejoras en Blazor

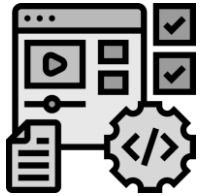
Mejoras en SignalR

APIs Ligeras con Minimal APIs

Nuevas herramientas para simplificar la depuración, incluidas mejoras en Visual Studio y soporte para depuración remota en aplicaciones contenerizadas. También incorpora diagnósticos más detallados para servicios distribuidos.

- **Detección más rápida de errores**: Herramientas visuales para identificar problemas en tiempo real.
- **Depuración remota**: Ideal para aplicaciones distribuidas en Kubernetes o Docker.
- **Mejor integración con CI/CD**: Automatiza pruebas y diagnósticos en pipelines.

Nuevas características de ASP.NET Core 9

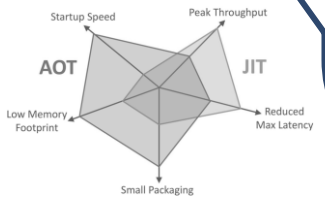


Optimización de Entrega de Recursos

Mejoras en Blazor

Mejoras en SignalR

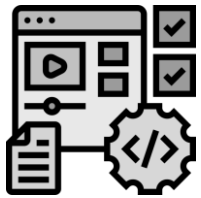
APIs Ligeras con Minimal APIs



La compilación Ahead-of-Time (AoT) genera aplicaciones más ligeras y rápidas al optimizar el código para escenarios específicos. Esto es especialmente útil para dispositivos de bajo consumo y entornos IoT.

- **Rendimiento superior**: Tiempos de inicio más rápidos y menor uso de memoria.
- **Tamaño reducido**: Binarios más pequeños, ideales para dispositivos embebidos.
- **Escenarios específicos**: Optimización dirigida a aplicaciones con requerimientos de alto rendimiento.

Nuevas características de ASP.NET Core 9



Optimización de Entrega de Recursos

Mejoras en Blazor

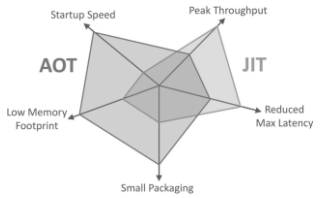
Mejoras en SignalR

APIs Ligeras con Minimal APIs

Documentación con OpenAPI

Autenticación y Autorización Mejoradas

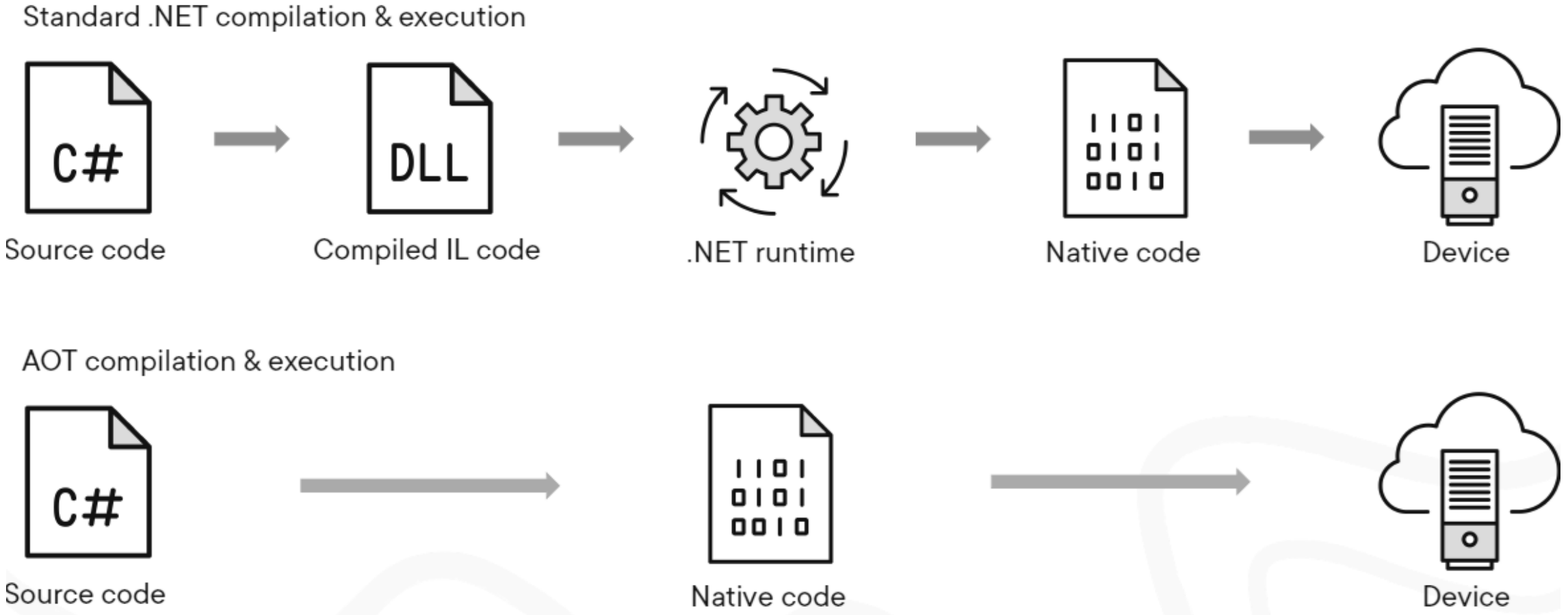
Mejoras en Depuración y Herramientas



Optimización con Ahead-of-Time (AoT)

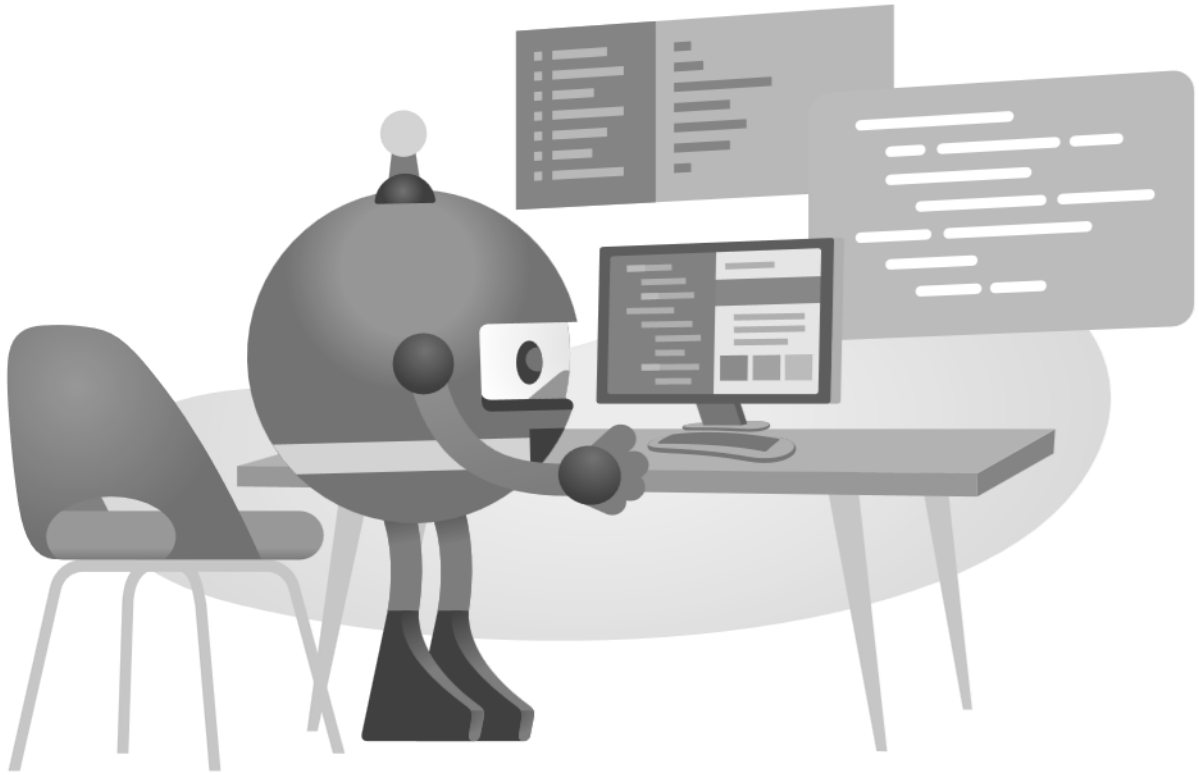
GALAXY TRAINING

NET Compilation Workflows



ASP.NET Core novedades y mejoras

DEMO



GALAXY
TRAINING

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)

GALAXY TRAINING

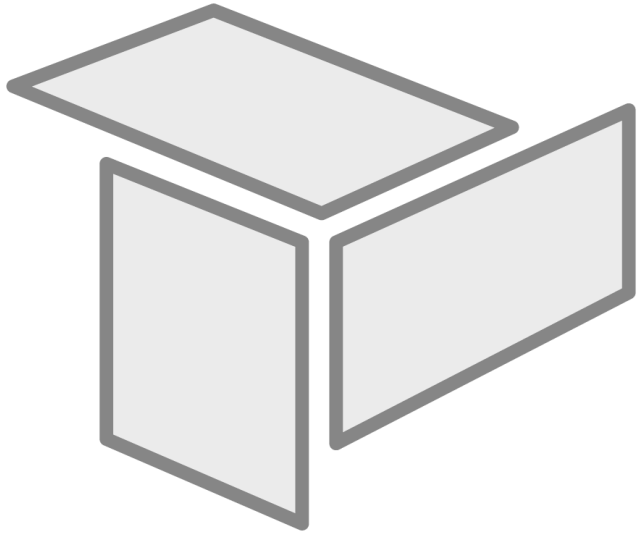
Plataforma madura - ASP.NET 8

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);
});
```

GALAXY TRAINING

Route groups - ASP.NET 8

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

GALAXY TRAINING

TypedResults - ASP.NET 8

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

Minimal APIs vs Controller APIs(Classic)

Característica	Minimal APIs	APIs Basadas en Controladores
Complejidad del Proyecto	Ideal para proyectos pequeños, microservicios o APIs específicas.	Mejor para proyectos grandes, estructurados o con requisitos avanzados.
Configuración Inicial	Baja: configuración rápida y código reducido.	Alta: requiere configuración detallada y más código.
Model Binding	No soportado nativamente; requiere implementación manual o soluciones personalizadas.	Soporte completo para IModelBinder y IModelBinderProvider.
Validación de Datos	No soportado nativamente (IModelValidator); validación manual o con bibliotecas externas.	Validación nativa a través de IModelValidator y DataAnnotations.
Rutas y Enrutamiento	Definición directa con lambdas o métodos; admite expresiones regulares en rutas.	Admite enrutamiento avanzado, con soporte nativo para Application Parts y convenciones globales.
Soporte de OpenAPI	Soporte mejorado en .NET 9; permite personalizar documentación y respuestas.	Totalmente integrado con documentación avanzada y soporte para herramientas externas como Swagger.
JSON y JsonPatch	No soportado; JsonPatch no está disponible nativamente.	Soporte completo para JsonPatch, ideal para actualizaciones parciales en JSON.

Minimal APIs vs Controller APIs(Classic)

Característica	Minimal APIs	APIs Basadas en Controladores
OData	No soportado; no es compatible con consultas avanzadas de OData.	Totalmente compatible con OData, ideal para consultas avanzadas en APIs RESTful.
Renderizado de Vistas	No soportado; enfocado solo en lógica de APIs.	Compatible con Razor Pages y vistas basadas en controladores.
Inyección de Dependencias	Soporte básico a través de HttpContext.RequestServices u otros métodos.	Soporte completo para inyección de dependencias mediante constructor o propiedades.
Estructura de Código	Código minimalista y funcional; menos modularidad para proyectos grandes.	Código modular y orientado a objetos, ideal para escalabilidad y mantenibilidad.
Desempeño	Optimizado para respuestas rápidas y uso ligero de recursos.	Similar, pero puede ser más lento debido a la configuración y validación adicional.
Facilidad de Pruebas	Pruebas directas para APIs pequeñas, pero limitaciones en escenarios complejos.	Mejor soporte para pruebas unitarias y de integración debido a su estructura más rica.

03

Arquitectura de Minimal APIs

GALAXY TRAINING

Opciones para estructurar Minimal APIs

GALAXY TRAINING

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

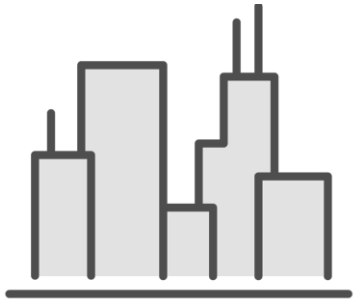
GALAXY TRAINING

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

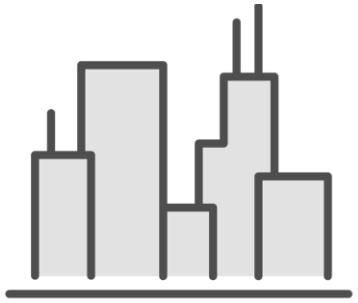
GALAXY TRAINING

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

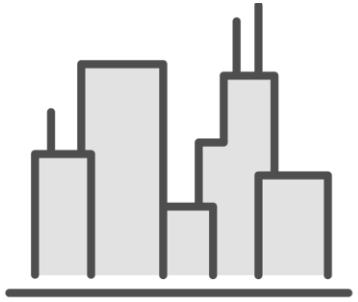
Librería de terceros

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

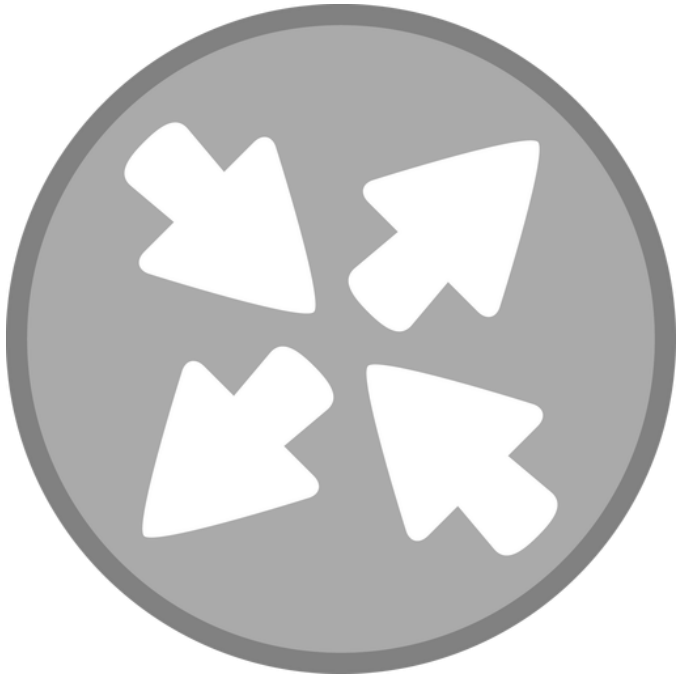
Application/WebApplicationBuilder y Routing

GALAXY TRAINING

Routing

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

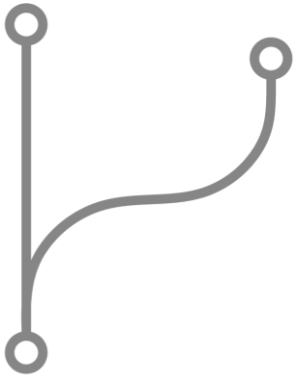
Routing



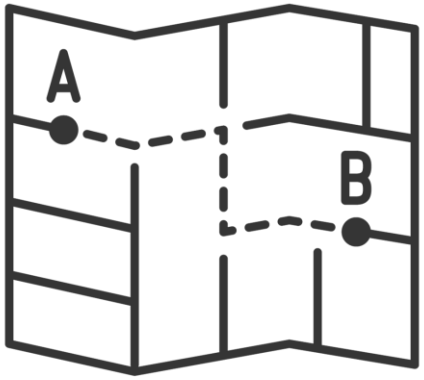
app.**Map*Action*** methods

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

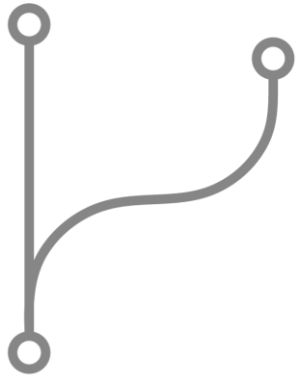
Conceptos de Routing



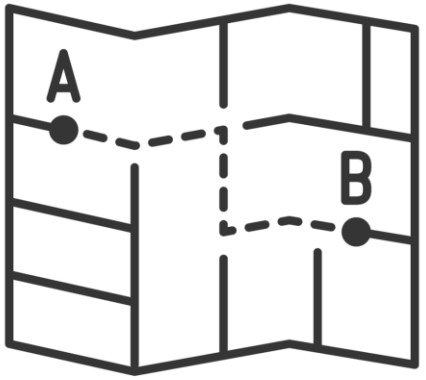
Metodo Route builder de **IEndpointRoute Builder**



Plantillas o patrones de Rutas



Parámetros de Rutas



Restricciones de Rutas

GALAXY TRAINING

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

GALAXY
TRAINING

GALAXY TRAINING

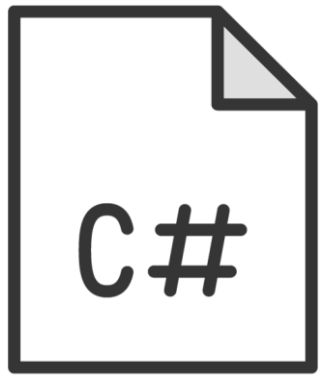
Routing

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 vacio
DELETE	**MapDelete**	-	/authors/{authorId}	-

Valores retornables de un Minimal API

string

Cualquier tipo, pero
se genera una cadena



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

GALAXY TRAINING

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

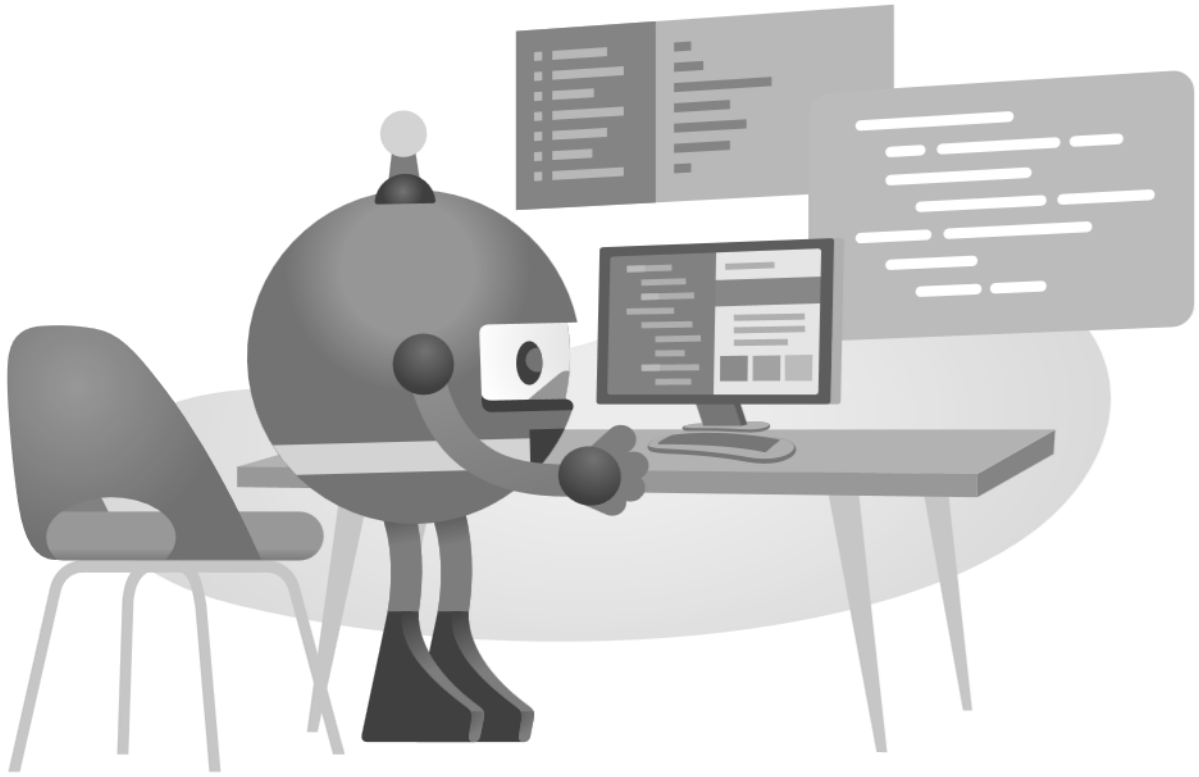
06

Desarrollando y desplegando pruebas de concepto(PoC)

GALAXY
TRAINING

Desarrollando y desplegando pruebas de concepto(PoC)

DEMO



GALAXY
TRAINING

GRACIAS

POR SU PREFERENCIA

GALAXY
TRAINING