# 📌 Parámetros en métodos HTTP (ASP.NET Core)

Cuando definís un método en un controlador, ASP.NET Core se encarga de mapear automáticamente los valores que viajan en la request hacia los parámetros del método. Este proceso se llama Model Binding.

## 🔹 ¿Cómo decide de dónde sacar el valor?

ASP.NET Core sigue un orden de búsqueda para enlazar parámetros. Si no lo indicás explícitamente, intenta resolverlos en este orden:

### 1. [FromRoute] → Ruta (URL con placeholders)

[HttpGet("search/{search}")]  
public IActionResult Get([FromRoute] string search)

👉 URL: /api/people/search/nico  
✅ Resultado: search = "nico"

### 2. [FromQuery] → Query string (?key=value)

[HttpGet("search")]  
public IActionResult Get([FromQuery] string search)

👉 URL: /api/people/search?search=nico  
✅ Resultado: search = "nico"

### 3. [FromBody] → Cuerpo de la request (POST, PUT, PATCH)

[HttpPost("search")]  
public IActionResult Post([FromBody] string search)

👉 Body JSON: { "search": "nico" }  
✅ Resultado: search = "nico"

### 4. [FromHeader] → Headers personalizados

public IActionResult Get([FromHeader(Name = "x-custom-header")] string value)

👉 Header: x-custom-header: hola  
✅ Resultado: value = "hola"

### 5. [FromForm] → Datos enviados en formularios (HTML form-data)

[HttpPost("upload")]  
public IActionResult Upload([FromForm] string name)

👉 FormData: name=nico  
✅ Resultado: name = "nico"

## ✅ Resumen rápido para recordar

- {param} en la ruta → [FromRoute] (URL segment).  
- ?key=value → [FromQuery] (query string).  
- JSON en body → [FromBody] (POST/PUT/PATCH).  
- Header HTTP → [FromHeader].  
- Formulario HTML → [FromForm].  
  
📌 Tip: Si no ponés atributo, ASP.NET Core intenta adivinar. Por ejemplo, los tipos simples (int, string) los busca en query o route, y los complejos (clases) en el body.