CONTROLADOR CON DAPPER y EF

// Archivo: Controllers/ProductosController.cs

using Dapper;

using Microsoft.Data.SqlClient;

using Microsoft.EntityFrameworkCore;

using System.Data;

[ApiController]

[Route("api/[controller]")]

public class ProductosController : ControllerBase

{

    private readonly ApplicationDbContext \_contexto;

    private readonly string \_cadenaConexion; // <-- Aquí se define

    // El constructor inyecta el DbContext (para EF) y la configuración (para Dapper)

    public ProductosController(ApplicationDbContext contexto, IConfiguration configuracion)

    {

        \_contexto = contexto;

        \_cadenaConexion = configuracion.GetConnectionString("DefaultConnection"); // <-- Aquí se inicializa

    }

    // Endpoint con EF (crear un producto)

    [HttpPost]

    public async Task<ActionResult<Producto>> PostProducto(Producto producto)

    {

        \_contexto.Productos.Add(producto);

        await \_contexto.SaveChangesAsync();

        return CreatedAtAction(nameof(GetProducto), new { id = producto.Id }, producto);

    }

    // Endpoint con Dapper (lectura optimizada)

    [HttpGet("listado")]

    public async Task<IActionResult> GetProductosListado()

    {

        var sql = "SELECT ..."; // Tu consulta SQL

        using (IDbConnection db = new SqlConnection(\_cadenaConexion))

        {

            var productos = await db.QueryAsync<ProductoListadoDTO>(sql);

            return Ok(productos);

        }

    }

}

EXTRAER POR NOMBRE  
[ApiController]

[Route("api/[controller]")]

public class ProductosController : ControllerBase

{

private readonly ApplicationDbContext \_contexto;

private readonly string \_cadenaConexion;

public ProductosController(ApplicationDbContext contexto, IConfiguration configuracion)

{

\_contexto = contexto;

\_cadenaConexion = configuracion.GetConnectionString("DefaultConnection");

}

// Endpoint para obtener todos los productos (con Dapper)

[HttpGet("listado")]

public async Task<IActionResult> GetProductosListado()

{

// ... (código que ya conoces)

}

// Nuevo endpoint para obtener televisores que empiezan con 'T'

[HttpGet("televisores/nombre/{nombre}")]

public async Task<IActionResult> GetTelevisoresPorNombre(string nombre)

{

// Aquí obtienes el ID de la categoría "Televisores"

// Esta consulta puede ser con EF o Dapper, dependiendo de tu preferencia

var idCategoriaTelevisores = \_contexto.Categorias.FirstOrDefault(c => c.Nombre == "Televisores")?.Id;

if (idCategoriaTelevisores == null)

{

return NotFound("Categoría 'Televisores' no encontrada.");

}

var sql = @"

SELECT p.Id, p.Nombre, p.Descripcion, p.Precio

FROM Productos p

WHERE p.CategoriaId = @CategoriaId

AND p.Nombre LIKE @Nombre";

using (var conexion = new SqlConnection(\_cadenaConexion))

{

var televisores = await conexion.QueryAsync<ProductoListadoDTO>(sql, new {

CategoriaId = idCategoriaTelevisores,

Nombre = nombre + "%"

});

return Ok(televisores);

}

}

}