

Universidad Técnica de Ambato

Nombre: Jesica Soria

Semestre: 3ro TI "A"

Fecha: 23/06/2025

Prueba: Programación Avanzada Introducción

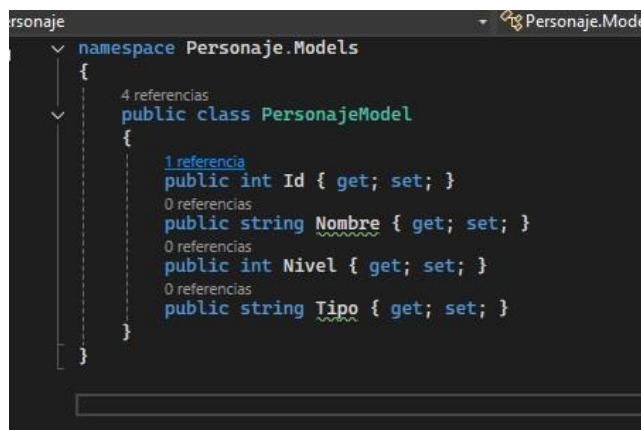
Este informe describe el desarrollo de una aplicación Web API construida con el framework ASP.NET Core, utilizando Entity Framework Core para el acceso a datos. El propósito fundamental del proyecto es administrar una entidad denominada Personaje, la cual representa un personaje dentro de un sistema, facilitando su almacenamiento, consulta e inserción en una base de datos SQL Server.

Herramientas y tecnologías

- Visual Studio 2022
- ASP.NET Core 6/7
- Entity Framework Core
- SQL Server (SQLEXPRESS)
- Swagger para pruebas de endpoints
- C# como lenguaje principal

Modelado de la Entidad

Se modeló una clase llamada PersonajeModel con las siguientes propiedades:

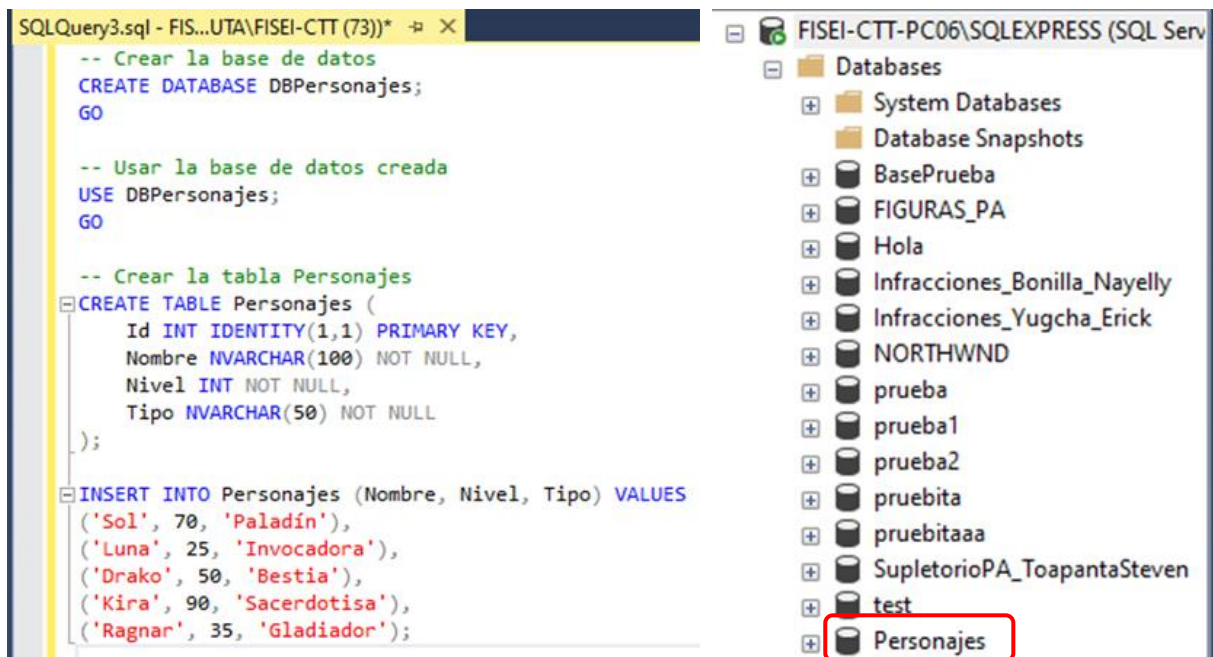


Esta clase se encuentra dentro del namespace Personaje.Models y se mapea a una tabla de base de datos mediante Entity Framework.

Configuración de la base de datos

Se creó la base de datos y la tabla mediante el siguiente script SQL

También se insertaron datos de prueba:

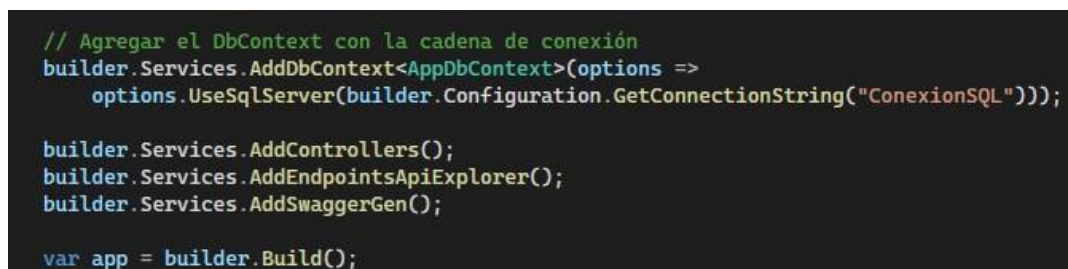


Configuración de la conexión a SQL Server

La cadena de conexión fue agregada en el archivo appsettings.json:



Y registrada en Program.cs:



Configuración del DbContext

```

using Personaje.Models;

namespace Personaje.Data
{
    5 referencias
    public class ApplicationDbContext : DbContext
    {
        0 referencias
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options) { }

        2 referencias
        public DbSet<PersonajeModel> Personajes { get; set; }
    }
}

```

Este contexto se encarga de gestionar la comunicación con la base de datos.

Desarrollo del controlador API

Se implementó el controlador PersonajesController.cs con los métodos básicos GET y POST:

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Personaje.Data;
using Personaje.Models;

namespace Personaje.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    1 referencia
    public class PersonajesController : ControllerBase
    {
        private readonly ApplicationDbContext _context;

        0 referencias
        public PersonajesController(ApplicationDbContext context)
        {
            _context = context;
        }

        // GET: api/personajes
        [HttpGet]
        1 referencia
        public async Task<ActionResult<IEnumerable<PersonajeModel>>> Get()
        {
            return await _context.Personajes.ToListAsync();
        }

        // POST: api/personajes
        [HttpPost]
        0 referencias
        public async Task<ActionResult<PersonajeModel>> Post(PersonajeModel personaje)
        {
            _context.Personajes.Add(personaje);
            await _context.SaveChangesAsync();

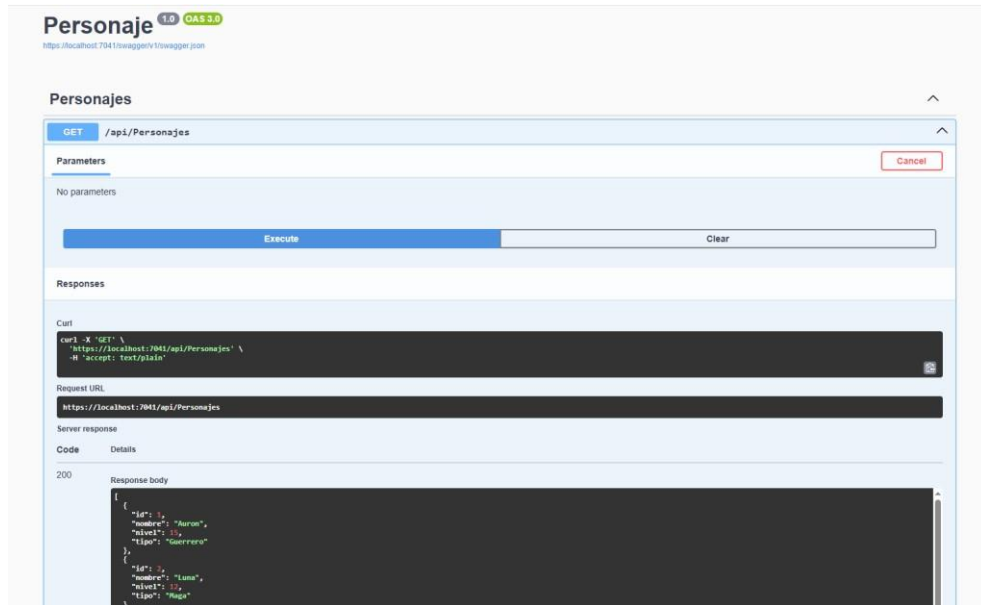
            return CreatedAtAction(nameof(Get), new { id = personaje.Id }, personaje);
        }
    }
}

```

Pruebas con Swagger

Se utilizó la herramienta **Swagger UI** (habilitada por defecto en el proyecto) para probar la funcionalidad de la API:

- GET /api/personajes muestra todos los personajes almacenados.
- POST /api/personajes permite insertar un nuevo personaje.



Conclusión

La creación de esta API evidenció la aplicación efectiva de ASP.NET Core en conjunto con Entity Framework para desarrollar una solución robusta, bien estructurada y mantenible. Se estableció exitosamente la conexión con SQL Server, se diseñó adecuadamente el modelo de datos y se expusieron endpoints REST operativos. Asimismo, las funcionalidades fueron validadas correctamente mediante pruebas con datos reales utilizando Swagger.

Este proyecto constituye un punto de partida ideal para desarrollar sistemas más avanzados que requieran persistencia de datos y una arquitectura organizada por capas.

Link GitHub: <https://github.com/Soria299/Personajes>