1. **Conexión a la base de datos:**

En appsettings.json :

"ConnectionStrings": {

"ConexionMontecastelo": "Data Source=.\\MSSQLSERVER01;Initial Catalog=Nombre de la base de datos;Trusted\_Connection=true;MultipleActiveResultSets=true;TrustServerCertificate=True"

}

1. **Añadir la conexión:**

En Program.css

builder.Services.AddDbContext<ProductContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("ConexionMontecastelo")));

1. **Se añade el contetxo de la conxion d e la base de datos :** crea una clase:

public class ProductContext: DbContext

{

public ProductContext (DbContextOptions< ProductContext > options) : base(options) { }

public DbSet<Producto> Products { get; set; }

}

1. **Creas los modelos igual q está en la base de datos :**

**[Table("Nombre de la tabla")]**

**public class** Producto

**{**

**[Key]**

**public int Id { get; set; }**

**public string Nombre { get; set; }**

**public decimal Precio {get; set; }**

**}**

1. **Crea una carpeta repositorio luego una clase EjmploReposiotio :**

public class Producto**Reposiotio**: IDisposable

{

private readonly EjemploContext \_context;

public Producto**Reposiotio** (ProductoContext context)

{

\_context = context;

}

**//Esto obtiene todos lo q tiene la tabla**

public IQueryable< Producto > GetAllProducto()

{

return \_context.Products.AsQueryable();

}

**// Mediante el id muestra los resultados**

public Producto GetProductoByID(int id)

{

return \_context.Products.FirstOrDefault(producto => producto.Id == id);

}

**//Elimina mediante id public void Delete(int id)**

public void Delete(int id)

{

var producto = GetProductoByID( id);

\_context.Products.Remove(producto);

\_context.SaveChanges();

}

**//Guarda los cambios**

public void Save(Producto producto)

{

\_context.Products.Add(producto);

\_context.SaveChanges();

}

**//Actualzia**

public void Update()

{

\_context.SaveChanges();

}

public void Dispose()

{

\_context.Dispose();

}

}

1. **Crea los controller necesario :**

public class EjemploController : Controller

{

private readonly EjemploRepositorio \_ejemploRepositorio;

public EjemploController (EjemploContext context)

{

\_ejmploRepositorio = new EjemploRepositorio (context);

}

[Route("/")]

[ Route("/ejemplo")]

[Route("/ejemplo/Index")]

public IActionResult Index()

**// Muestra la lista**

{ var ejemplos = \_ ejmploRepositorio.GetAllEjemplo();

return View(ejemplos);

}

}

**//Muestra mediante el id**

public IActionResult Detalle(int id)

{

var productos = \_productoRepositorio.GetProductoByID(id);

return View(productos);

}

**//Agrega**

[Route("/Producto/agregar")]

public IActionResult Agregar()

{

return View(new Producto());

}

**//Crea el producto**

[HttpPost]

[Route("/Producto/create")]

public IActionResult Create()

{

var Id = int.Parse(HttpContext.Request.Form["Id"]);

var nombre = HttpContext.Request.Form["Nombre"];

var precio = decimal.Parse(HttpContext.Request.Form["Precio"].ToString().Replace(".", ","));

var productos = \_productoRepositorio.GetProductoByID(Id);

if (productos == null)

{

\_productoRepositorio.Save(new Producto

{

Id = Id,

Nombre = nombre,

Precio = precio,

}); ;

}

else

{

productos.Nombre = nombre;

productos.Precio = precio;

\_productoRepositorio.Update();

}

var productost = \_productoRepositorio.GetAllProducto();

return View("Index", productost);

}

**//Elimina el producto mediante id**

[Route("/Producto/delete/{Id}")]

public IActionResult Delete(int id)

{

\_productoRepositorio.Delete(id);

var pokemons = \_productoRepositorio.GetAllProducto();

return View("Index", pokemons);

}

**//Edita el producto mediante el id**

[Route("/Producto/edit/{Id}")]

public IActionResult Edit(int Id)

{

var producto = \_productoRepositorio.GetProductoByID(Id);

return View("Agregar", producto);

}