proyecto con ASP.NET

**Primer mandato**

Crea una API usando asp.net Core

El api es para un sistema de inventario, donde tendrás productos, movimientos (entrada y salida)

Y los datos debes persistirlos/guardarlos en una base de datos

estructura de base de datos

Tipos de inventario de Utilizado: Inventario Periódico

Base de datos Stocktaking\_JK

**Según mandato**

1. Se para el proyecto en capas. Formalmente se le llama N-tier Architecture.

2. Limpia el proyecto eliminando los elementos que no son necesarios (WeatherForecastController, por ejemplo). Esto incluye solo versionar los elementos mínimos necesarios para que la aplicación funcione (usa. gitignore)

**1- Productos**

ID del producto (clave primaria)

Nombre del producto

Descripción del producto

Precio del producto

Cantidad Producto

ID de la categoría (clave foránea)

**2- categoría**

ID de la categoría (clave primaria)

Nombre de la categoría

3- Movimiento

ID del movimiento (clave primaria)

ID del producto (clave foránea)

Tipo de movimiento (entrada o salida)

Cantidad

Fecha y hora del movimiento

**4- proveedor**

ID del proveedor (clave primaria)

Nombre del proveedor

Dirección del proveedor

Contacto del proveedor

**5- Región**

ID de la región (clave primaria)

Nombre de la región

Dirección de la región

6- Adquisición

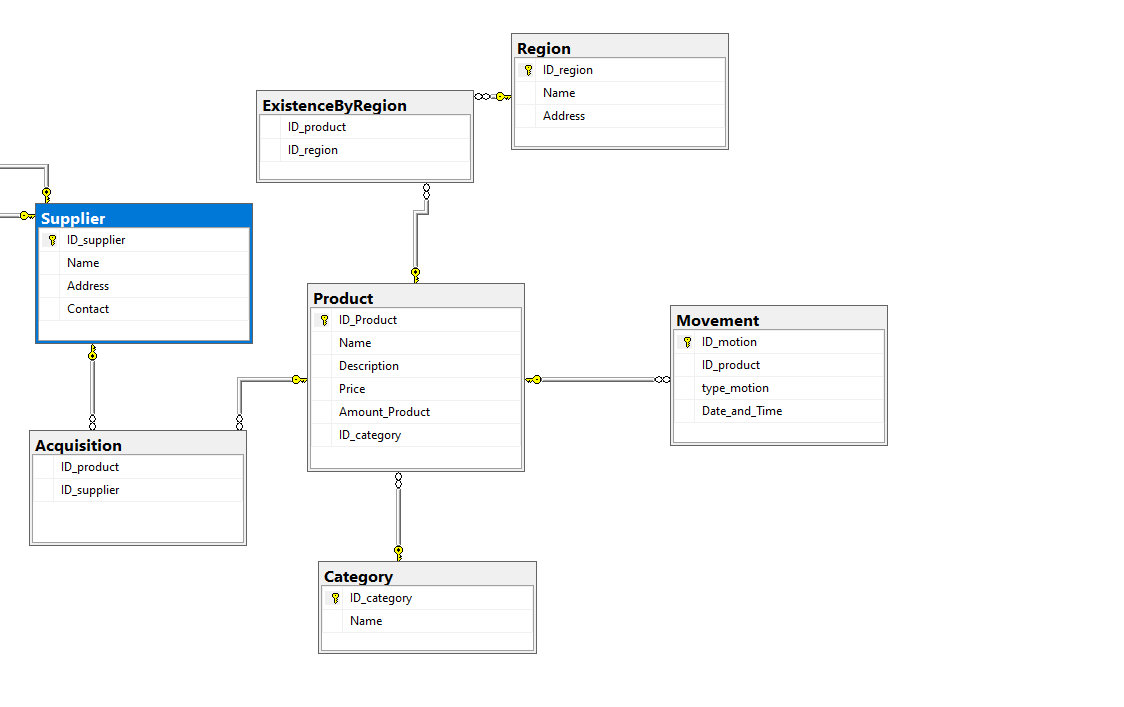
ID del producto (clave foránea)

ID del proveedor (clave foránea)

**7- ExistenciasPorRegion**

ID del producto (clave foránea)

ID de la región (clave foránea)



Entity framework

Sever= DESKTOP-CSJVP05\SQLEXPRESS

Paquete = Microsoft.EntityFrameworkCore.SqlServer

Errors y solutions

**Error**

Keyword not supported:

**Solutions**

Error de escritura

**Error**

Base de datos no es confiable

**Solución**

“TrustServerCerificate=True”

**Nota**

* Database First

Conexión y creación de clase por método Database First

**Scaffold-DbContext "Server=DESKTOP-CSJVP05\SQLEXPRESS; Database=Stocktaking\_JK; Trusted\_Connection=True; TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models**

Actualización de cambios en la base de datos

* Code First

Secuencia de cadena de conexión para hacer la conexión en json pueda ser utilizada en otros lenguaje

"ConnectionStrings": {

"StockContext": "Scaffold-DbContext Server=DESKTOP-CSJVP05\\SQLEXPRESS; Database=Stocktaking\_JK; Trusted\_Connection=True; TrustServerCertificate=True;"

}

Comando nugget paquete

Conexión y creación de clase por método Database First

**Scaffold-DbContext "Server=DESKTOP-CSJVP05\SQLEXPRESS; Database=Stocktaking\_JK; Trusted\_Connection=True; TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models**

Estandarización HttpGet para leer

[HttpGet ("{ID\_product}")]

public async Task<ActionResult<Product>> GetProducts( int ID\_product )

{

if (\_dbcontext.Products == null)

return NotFound();

var Product = await \_dbcontext.Products.FindAsync(ID\_product );

if (Product == null)

return NotFound();

return Product;

}