**Implementación Patrón de Repositorio (Repository Pattern)**

Deberás implementar el patrón de repositorio para el modelo producto que ya está en el ejercicio.

Crea un interfaz con los métodos, puedes usar los nombres de la derecha.

Despues deberás crear la clase o servicio c# que implementa los métodos de la interfaz anterior. Aquí debes uar Entity Framework Core con los métodos básicos vistos en las claes anteriores, ejemplo: Add, Find

Recuerda usar las convenciones mencionadas en el curso, por ejemplo para las interfaces usamos la "I" en mayuscula al principio.

Objetivo de aprendizaje

Este tema aparece en

using System.Collections.Generic;

namespace Coding.Exercise

{

public class Producto

{

//Aquí esta el modelo sobre el que trabajaremos. Deberas implemetnar el Patrón de Repositorio (Repository Pattern)

//para este modeo

public int ProductoId { get; set; }

public string Nombre { get; set; }

public decimal Precio { get; set; }

}

//Escribe aquí la interfaz con los métodos para realizar solamente 3 operaciones para el modelo producto:

public interface IProductoRepository

{

//3 AgregarProducto

Producto AgregarProducto(Producto producto);

//2-ObtenerProductoPorId: Permite obtener un solo producto por su Id

Producto ObtenerProductoPorId(int productoId);

//1-ObtenerTodosLosProductos: Permite obtener todos los productos

List<Producto> ObtenerTodosLosProductos();

}

//3. Implementación del Repositorio:

public class ProductoRepository : IProductoRepository

{

private readonly List<Producto> \_productos = new List<Producto>();

private int \_productoIdCounter = 1;

public Producto AgregarProducto(Producto producto)

{

producto.ProductoId = \_productoIdCounter++;

\_productos.Add(producto);

return producto;

}

public Producto ObtenerProductoPorId(int productoId)

{

return \_productos.Find(p => p.ProductoId == productoId);

}

public List<Producto> ObtenerTodosLosProductos()

{

return \_productos;

}

}

}