**Práctica 01**

**Antes de realizar la práctica leer**

**Clase, objeto y propiedades.docx en el recurso compartido**

**Introducción**

En esta fase comenzaremos a crear algunos de los objetos que gestionará nuestra aplicación.

Estos objetos, a los que llamaremos entidades, funcionarán como si fuesen tipos de datos personalizados. Los objetos a codificar por el momento serán Producto, Proveedor, Cliente, Vendedor, Factura y Remito.

Habilidades y conocimientos aplicados en esta fase

* Crear soluciones y proyectos en Visual Studio.
* Comprender la arquitectura basada en capas.
* Crear clases.
* Codificar atributos de lectura/escritura como propiedades autoimplementadas.
* Codificar atributos de sólo lectura.
* Comprender el concepto de visibilidad.
* Crear constructores paremetrizados.
* Crear referencias e importar espacios de nombres.
* Instanciar objetos.

Paso 1

* Crear una solución en tipo consola: **slnCursoNet**
* Agregar un nuevo proyecto del tipo **Consola** a la solución.
* Nombre: **ClaseDemoApp**.

Paso 2

* Agregar a la solución un proyecto del tipo librería de clases y llamarlo Entidades.

Paso 3

* Dentro del proyecto Entidades, crear las clases que se detallan a continuación y codificar sus atributos como propiedades autoimplementadas, con excepción de los atributos de sólo lectura, en los cuales deberá codificarse el método Get().

**Producto**

1. Nombre string lectura/escritura
2. Descripcion string lectura/escritura
3. PrecioCosto decimal lectura/escritura
4. Margen double lectura/escritura
5. IVA double lectura/escritura
6. PrecioBruto decimal sólo lectura
7. PrecioVenta decimal sólo lectura
8. Proveedor string lectura/escritura
9. Categoria string lectura/escritura
10. SubCategoria string lectura/escritura

**Proveedor**

1. Nombre string lectura/escritura
2. CUIT string lectura/escritura
3. Contacto string lectura/escritura
4. Email string lectura/escritura
5. Telefono string lectura/escritura
6. Direccion string lectura/escritura

**ClienteIndividuo**

1. Nombre string lectura/escrituta
2. Apellido string lectura/escritura
3. CUIT string lectura/escritura
4. Email string lectura/escritura
5. Telefono string lectura/escritura
6. Direccion string lectura/escritura

**ClienteEmpresa**

1. Nombre string lectura/escrituta
2. CUIT string lectura/escritura
3. Contacto string lectura/escritura
4. Email string lectura/escritura
5. Telefono string lectura/escritura
6. Direccion string lectura/escritura

**Vendedor**

1. Nombre string lectura/escritura
2. Apellido string lectura/escritura
3. DNI string lectura/escritura
4. Email string lectura/escritura
5. Telefono string lectura/escritura
6. Direccion string lectura/escritura

**Factura**

1. Tipo string lectura/escritura
2. Numero string lectura/escritura
3. Fecha date lectura/escritura
4. Cliente string lectura/escritura
5. Direccion string lectura/escritura
6. CondicionIVA string lectura/escritura
7. CondicionVenta string lectura/escritura
8. Detalle string lectura/escritura
9. Total decimal lectura/escritura

**Remito**

1. Numero string lectura/escritura
2. Fecha date lectura/escritura
3. Cliente string lectura/escritura
4. Direccion string lectura/escritura
5. CondicionIVA string lectura/escritura
6. CondicionVenta string lectura/escritura
7. Detalle string lectura/escritura
8. FechaEntrega date lectura/escritura
9. Total decimal lectura/escritura

Paso 5

Crear una referencia desde Consola a Entidades. Importar el espacio de nombres Entidades en Consola.

Paso 6

Dentro del proyecto Consola, instanciar las clases, asignarles valores a sus atributos y mostrar datos por pantalla.

Dadas las siguientes clases.

public class Book

{

string Title { get; set; }

int AuthorId { get; set; }

int PublicationDate { get; set; } // Year

int Sales { get; set; } //Millions

public Book(string title, int authorId, int publicationDate, int sales)

{

this.Title = title;

this.AuthorId = authorId;

this.PublicationDate = publicationDate;

this.Sales = sales;

}

public static List<Book> Books()

{

return new List<Book>

{

new Book("Don Quijote de la Mancha", 1, 1605, 500),

new Book("Historia de dos ciudades", 2, 1859, 200),

new Book("El Señor de los Anillos", 3, 1978, 150),

new Book("El principito", 4, 1951, 140),

new Book("El hobbit", 3, 1982, 100),

new Book("Sueño en el pabellón rojo", 5, 1792, 100),

new Book("Las aventuras de Alicia en el país de las maravillas", 6, 1865, 100),

new Book("Diez negritos", 7, 1939, 100),

new Book("El león, la bruja y el armario", 8, 1950, 85),

new Book("El código Da Vinci", 9, 2003, 80),

new Book("El guardián entre el centeno", 10, 1951, 65),

new Book("El alquimista", 11, 1988, 65),

};

}

}

public class Author

{

public int AuthorId { get; set; }

public string Name { get; set; }

public Author(int authorId, string name)

{

this.AuthorId = authorId;

this.Name = name;

}

public static List<Author> Books()

{

return new List<Author>

{

new Author(1, "Miguel de Cervantes"),

new Author(2, "Charles Dickens"),

new Author(3, "J. R. R. Tolkien"),

new Author(4, "Antoine de Saint-Exupéry"),

new Author(5, "Cao Xueqin"),

new Author(6, "Lewis Car"),

new Author(7, "Agatha Christie"),

new Author(8, "C. S. Lewis"),

new Author(9, "Dan Brown"),

new Author(10, "J. D. Salinger"),

};

}

}

1.- Mostrar en consola los 3 libros con más ventas.

2.- Mostrar en consola los 3 libros con menos ventas.

4.- Mostrar en consola el autor con más libros publicados.

5.- Mostrar en consola el autor y la cantidad de libros publicados.

6.- Mostrar en consola los libros publicados hace menos de 50 años.

7.- Mostrar en consola el libro más viejo.

8.- Mostrar en consola los autores que tengan un libro que comience con "El".