**EXAMEN MODULO 1**

**Ejercicio 1 - Encapsulamiento y Herencia:**

**Enunciado**

Crea una aplicación en C# que modele un sistema de gestión para una biblioteca. En este sistema:

1. **Clases y Encapsulamiento:**
   * Crea una clase base Libro con las siguientes propiedades encapsuladas:
     + Titulo (string, lectura y escritura)
     + Autor (string, lectura y escritura)
     + ISBN (string, solo lectura, asignado al crear el libro).
     + CopiasDisponibles (int, lectura y escritura).
   * Crea métodos para:
     + Mostrar los detalles del libro (MostrarDetalles).
     + Reducir las copias disponibles en 1 (llamado PrestarLibro), este método debe ser virtual.
     + Aumentar las copias disponibles en 1 (llamado DevolverLibro).
2. **Herencia:**
   * Crea una clase LibroDigital que herede de Libro e incluya una propiedad adicional llamada URLDescarga (string, lectura y escritura). Este tipo de libro no debería permitir la reducción de copias disponibles (sobrescribe el método PrestarLibro para lanzar un mensaje informando que no aplica).
3. **Colecciones y Métodos Estáticos:**
   * Utiliza una lista (List<Libro>) para gestionar los libros disponibles.
   * Implementa un método estático en una clase estatica Biblioteca para **buscar un libro por su título** y **devolver sus detalles(MostrarEstadisticas)**
4. **Estructuras y Arreglos:**
   * Incluye una funcionalidad dentro de (**MostrarEstadisticas**) para mostrar las estadísticas de la biblioteca usando un arreglo o lista que indique:
     + Total de libros físicos.
     + Total de libros digitales.
     + Total de copias disponibles.

**Entrada esperada:**

* Crear instancias de libros físicos y digitales.
* Agregar estos libros a la lista de la biblioteca.
* Prestar y devolver libros, y mostrar las estadísticas.

**Ejercicio 2 - Polimorfismo y Herencia:**

Crea un programa en C# que administre un sistema de registro de vehículos. El programa debe demostrar el uso de **polimorfismo** y **herencia**, además de integrar los conceptos previamente estudiados.

**Requerimientos del Sistema**

1. **Clases y Objetos**:
   * Define una clase base Vehiculo con las siguientes propiedades protegidas:
     + Marca (string)
     + Modelo (string)
     + Año (entero)
     + PrecioBase (decimal)

Incluye un método virtual CalcularPrecioFinal() que calcule el precio final del vehículo (de manera básica será igual al PrecioBase).

1. **Herencia y Polimorfismo**:
   * Crea las clases VehiculoElectrico y VehiculoCombustion que hereden de Vehiculo.
   * En la clase VehiculoElectrico, sobrescribe CalcularPrecioFinal() para sumar un recargo por batería de $2000.
   * En la clase VehiculoCombustion, sobrescribe CalcularPrecioFinal() para agregar un impuesto del 10% al PrecioBase.
2. **Funciones y Procedimientos**:
   * Implementa un método estático en la clase Vehiculo que permita mostrar estadísticas de los vehículos registrados:
     + Total de vehículos.
     + Promedio de precios finales.
3. **Control de Flujo**:
   * Usa if-else o switch-case para decidir si un vehículo es eléctrico o de combustión al registrarlo.
   * Usa ciclos for, while, o do-while para recorrer el Menu interactivo
4. **Colecciones y Manejo de Datos**:
   * Utiliza una lista para almacenar los vehículos.
   * Utiliza un diccionario para contar la cantidad de vehículos por tipo ("Electrico" y "Combustion").

**Tareas para los Estudiantes**

1. Implementar las clases Vehiculo, VehiculoElectrico, y VehiculoCombustion con sus métodos y propiedades.
2. Registrar vehículos en una lista, solicitando los datos al usuario.
3. Implementar el método para mostrar estadísticas generales.
4. Mostrar todos los vehículos registrados, especificando el tipo y el precio final calculado.