

**Resumen**

Se presenta una simulación de una automotora mediante el uso de una aplicación web desarrollada en C# utilizando ASP.NET. Para gestionar la automotora tenemos clientes, vehículos, usuarios, ventas, alquileres y bases de datos (utilizando listas). Cuando se abre la aplicación, lo primero que se muestra es el login para ingreso de usuario que dependiendo que usuario sea, va tener permisos diferentes como, por ejemplo, el admin puede ver usuarios, clientes, vehículos, ventas y alquileres, así mismo puede dar alta, baja, modificar, actualizar y borrar. Al contrario de un vendedor, solo tiene permisos para listar y vender.

**Contenido**

[**1. Introducción** 1](#_Toc152509469)

[**1.1 .NET** 1](#_Toc152509470)

[**1.2 ASP.NET** 1](#_Toc152509471)

[**1.3 Visual Studio** 1](#_Toc152509472)

[**1.4 C#** 2](#_Toc152509473)

[**1.5 Listas** 2](#_Toc152509474)

[**1.6 UML** 2](#_Toc152509475)

[**2. Desarrollo** 2](#_Toc152509476)

[**2.1 UML de Obligatorio2023Prog2** 2](#_Toc152509477)

[**3. Resultados** 3](#_Toc152509478)

[**4. Conclusiones** 3](#_Toc152509479)

[**5. Referencias** 4](#_Toc152509480)

# **1. Introducción**

## **1.1 .NET**

.NET es una plataforma de código abierto que facilita la creación de aplicaciones para escritorio, web y móviles, con capacidad de ejecutarse en diversos sistemas operativos. La plataforma abarca herramientas, bibliotecas y lenguajes que respaldan el desarrollo de software moderno, escalable y de alto rendimiento. Respaldada por una comunidad de desarrolladores activa, .NET se mantiene y evoluciona constantemente. En resumen, .NET realiza varias funciones esenciales: Compilación de Código: Traduce el código del lenguaje de programación .NET en instrucciones comprensibles para los dispositivos de computación. Utilidades para Desarrollo Eficiente: Ofrece utilidades que facilitan el desarrollo de software eficiente, como la obtención de la hora actual o la impresión de texto en pantalla. Gestión de Tipos de Datos: Define un conjunto de tipos de datos para almacenar información, como texto, números y fechas en los equipos. Esta funcionalidad es crucial para la manipulación y almacenamiento de datos en las aplicaciones desarrolladas con .NET. [1]

## **1.2 ASP.NET**

Es un marco web gratuito que permite crear sitios web y aplicaciones web utilizando HTML, CSS y JavaScript. También posibilita la creación de API web y la integración de tecnologías en tiempo real, como Web Sockets. ASP.NET Core es una alternativa a ASP.NET, y se brinda orientación sobre cómo elegir entre ambas. Para comenzar, se recomienda instalar Visual Studio Community Edition, un entorno de desarrollo gratuito para ASP.NET en sistemas Windows. Existen tres marcos principales dentro de ASP.NET para crear aplicaciones web: Web Forms, ASP.NET MVC y ASP.NET Web Pages. Cada uno tiene su propio estilo de desarrollo y se adapta a diferentes niveles de experiencia y preferencias de programación. Web Forms: Ideal para aquellos con experiencia en Win Forms, WPF, o .NET. Ofrece un desarrollo rápido mediante una biblioteca de controles que encapsulan el marcado HTML. MVC: Recomendado para quienes tienen experiencia en Ruby on Rails o .NET. Proporciona control total sobre el marcado HTML, separación clara entre código y marcado, y es adecuado para aplicaciones móviles y de página única (SPA). Web Pages: Indicado para aquellos familiarizados con ASP clásico o PHP. Permite combinar código de servidor con HTML de manera rápida y ligera. Es importante destacar que los tres marcos comparten la funcionalidad principal de .NET y ASP.NET, como el modelo de seguridad de inicio de sesión. Además, no son mutuamente excluyentes y se pueden usar en la misma aplicación web. Además de estos marcos, ASP.NET ofrece opciones para la creación de API web con ASP.NET Web API y la implementación de funcionalidades en tiempo real con ASP.NET SignalR. En cuanto al desarrollo de sitios y aplicaciones móviles, ASP.NET puede potenciar aplicaciones nativas con un back-end de API web y sitios web móviles mediante marcos de diseño como Twitter Bootstrap. También se destaca la capacidad de desarrollar aplicaciones de página única (SPA) con HTML5, CSS3 y JavaScript, con plantillas disponibles en Visual Studio. ASP.NET proporciona una variedad de opciones y herramientas para el desarrollo web, adaptándose a diferentes necesidades y niveles de experiencia. [2]

## **1.3 Visual Studio**

Visual Studio destaca como una poderosa herramienta de desarrollo que facilita la gestión integral del ciclo de desarrollo en un único entorno. Este entorno de desarrollo integrado (IDE) ofrece capacidades completas para escribir, editar, depurar y compilar código, además de permitir la implementación de la aplicación resultante. Más allá de las funciones básicas de edición y depuración de código, Visual Studio proporciona una gama de herramientas y características adicionales. Entre ellas se incluyen compiladores, utilidades de finalización de código, control de código fuente, extensiones y diversas funcionalidades diseñadas para optimizar cada etapa del proceso de desarrollo de software. En síntesis, Visual Studio se presenta como una solución integral que va más allá de la simple edición de código, brindando a los desarrolladores un conjunto completo de recursos para mejorar la eficiencia y la calidad en todas las fases del desarrollo de aplicaciones. [3]

## **1.4 C#**

C# (pronunciado C Sharp) representa una evolución significativa realizada por Microsoft, amalgamando lo más destacado de los lenguajes C y C++. A lo largo de su desarrollo continuo, se le han incorporado funcionalidades provenientes de otros lenguajes, como Java, aprovechando aspectos de su sintaxis evolucionada. Este lenguaje, orientado a objetos en toda la plataforma NET (tanto Framework como Core), ha ido adquiriendo las facilidades de creación de código presentes en Visual Basic, otro de los lenguajes eminentemente utilizado por Microsoft. Esta amalgama le confiere a C# una versatilidad excepcional, convirtiéndolo en un lenguaje accesible y fácil de aprender, sin sacrificar la potencia inherente a C. Con la llegada de la versión .NET Core, se ha llevado a cabo una reconstrucción completa del compilador de C#, resultando en una mejora impresionante: las aplicaciones ahora se ejecutan hasta un 600% más rápido que en versiones anteriores. Este avance sustancial refuerza la posición de C# como un lenguaje moderno, potente y eficiente dentro del ecosistema de desarrollo de Microsoft. [4]

## **1.5 Listas**

En el contexto de C#, una lista (List) representa una estructura de datos diseñada para almacenar una colección de elementos del mismo tipo en un orden secuencial. A diferencia de los arrays, las listas carecen de un tamaño fijo, lo que permite la adición o eliminación dinámica de elementos. Adicionalmente, estas listas ofrecen una variedad de métodos que facilitan la manipulación de sus elementos. Entre las operaciones disponibles se encuentran la adición, eliminación, búsqueda, ordenamiento, y otras funcionalidades que optimizan la gestión de la información contenida en la lista. [5]

## **1.6 UML**

El Lenguaje de Modelado Unificado (UML) se erige como un estándar esencial para la representación visual de objetos, estados y procesos en el seno de un sistema. Desde una perspectiva, este lenguaje de modelado cumple la función de constituir un modelo para un proyecto, asegurando así una arquitectura de información sólidamente estructurada. Desde otro ángulo, UML facilita a los desarrolladores la tarea de presentar la descripción del sistema de una manera que resulte comprensible para aquellos que se encuentran fuera del ámbito especializado. En esencia, UML se emplea primordialmente en el desarrollo de software orientado a objetos. [6]

# **2. Desarrollo**

## **2.1 UML de Obligatorio2023Prog2**

El diagrama UML representa la estructura y las relaciones entre las clases en el sistema "Obligatorio2023Prog2". En este contexto, se encuentran las siguientes clases:

**BaseDeDatos:** Esta clase sirve como la base central del sistema, albergando listas de vehículos, clientes, usuarios, ventas y alquileres. Además, realiza un seguimiento del usuario que está autenticado en el sistema.

Usuario: La clase Usuario modela los usuarios del sistema, quienes pueden ser administradores o empleados. Cada usuario tiene permisos específicos para visualizar información relacionada con clientes, usuarios, vehículos, ventas y alquileres.

**Cliente:** Representa a los clientes que interactúan con el sistema. Almacena información como nombre, apellido, cédula y dirección. La clase Cliente también proporciona una función para validar el último dígito de la cédula uruguaya.

**Vehiculo:** Es la clase base para los vehículos y contiene atributos compartidos como marca, modelo, matrícula, etc. Además, existen clases derivadas específicas para Camiones, Motos y Autos.

**Camion, Moto y Auto:** Estas clases derivadas de Vehiculo representan tipos específicos de vehículos con atributos adicionales, como carga para camiones, cilindradas para motos y cantidad de pasajeros para autos.

**Alquiler y Venta:** Representan transacciones en las que se alquila o vende un vehículo a un cliente. Contienen información como la cédula del cliente, matrícula del vehículo, nombre de usuario del empleado, fecha y precio.

El diagrama también destaca las asociaciones entre estas clases, indicando las cardinalidades de las relaciones, como 1 a \*, 0 a 1 y 1 a 1. Estas relaciones describen cómo las instancias de las clases están conectadas entre sí dentro del sistema. (Figura 2.1)

# **3. Resultados**

# **4. Conclusiones**

# **5. Referencias**

[1] https://aws.amazon.com/es/what-is/net/

[2] <https://learn.microsoft.com/es-es/aspnet/overview>

[3] https://learn.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2022

[4] <https://bsw.es/que-es-c/>

[5] https://oregoom.com/c-sharp/listas/

[6] https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/