INTERFACES GRÁFICAS DE USUARIO

TRABAJO FINAL

Trabajo realizado por:

Juan Gil Sancho

Índice

1. Manual de usuario

1.1. Introducción

En este trabajo se ha desarrollado una aplicación para Windows utilizando el framework .NET y el lenguaje de programación C# junto al lenguaje de marcado XAML.

La aplicación consiste en un sistema de gestión de vehículos, en el cual se puede introducir datos de coches y de los repostajes que han ido haciendo a lo largo del tiempo y esta aplicación los almacenará y sacará estadísticas de los mismos referentes al número de kilómetros recorridos, una media del gasto en litros por cada 100 km y una media del gasto de cada repostaje también por cada 100 km.

Los datos se representarán en dos tipos de gráficos. La primera representación se realizará agrupando a todos los vehículos almacenados, en el que se compararán las medias mencionadas anteriormente y los kilómetros recorridos. La segunda representación es individual de cada vehículo y mostrará la evolución del gasto, consumo y kilómetros recorridos.

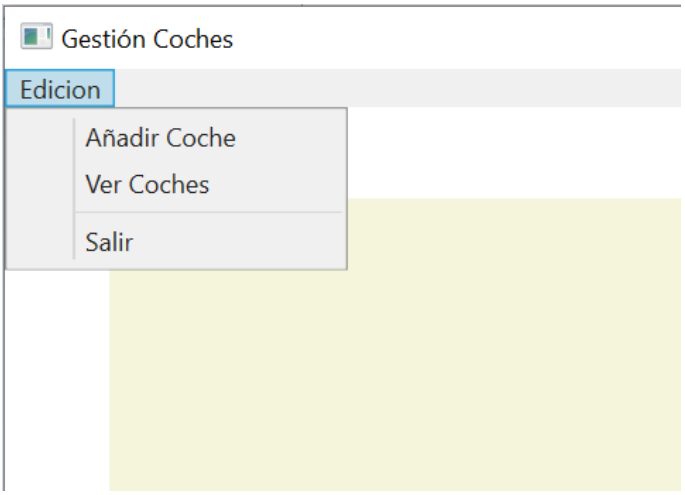
1.2. Funciones de la aplicación

En este apartado se describirán las funciones principales que tiene la aplicación a nivel del usuario.

1.2.1. Pantalla principal

En la siguiente imagen se muestra la pantalla principal de la aplicación.

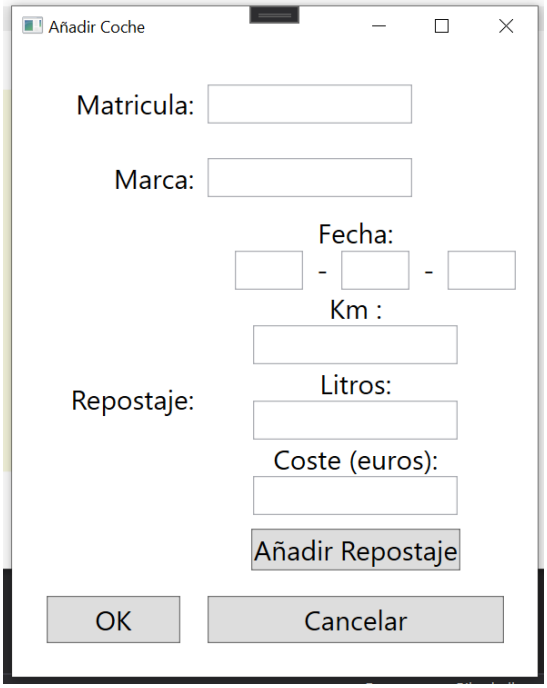
La pestaña de edición, como viene indicado en el documento, lleva a un menú en el que se muestran las distintas funciones que tiene la aplicación. Este menú se detallará en la siguiente imagen.



Las funciones que permite realizar la aplicación son añadir un coche o ver los coches registrados en la aplicación.

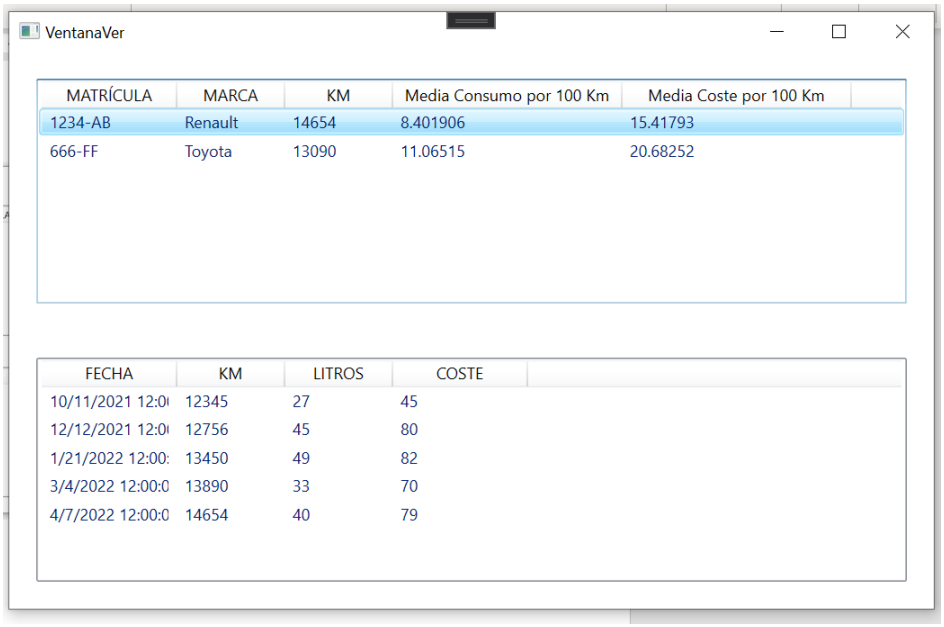
1.2.2. Añadir coche

La siguiente ventana permite añadir un coche al sistema. Para añadir un coche será necesario que tenga una matrícula, una marca y al menos un repostaje, el cual tiene que tener obligatoriamente fecha, kilómetros en el cuentakilómetros, litros y el coste en euros. Se pueden añadir tantos repostajes como se quieran.



1.2.3. Ver Coches

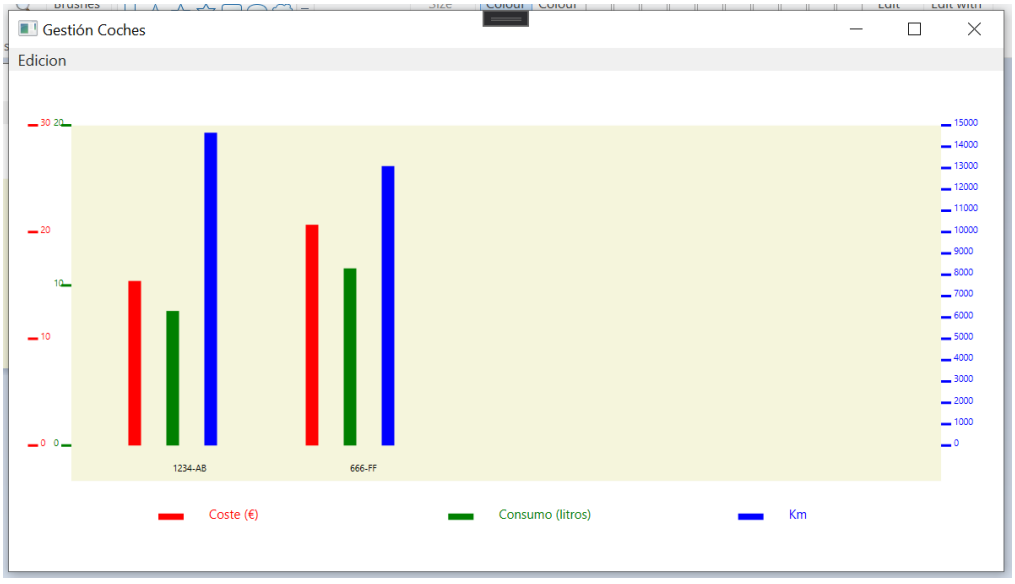
En la siguientes ventanas se muestran los coches registrados en el sistema en la primera lista, y en la segunda muestra los repostajes del coche seleccionado.



1.2.4. Gráficos

En este apartado se mostrarán los distintos tipos de gráficos estadísticos que muestra la aplicación.

1.2.4.1. Gráficos generales



En la imagen anterior se muestran un gráfico de líneas en el que se muestran los kilómetros del cuentakilómetros, la media de coste por cada 100 km y la media de consumo en litros por cada 100 km de todos los coches registrados.

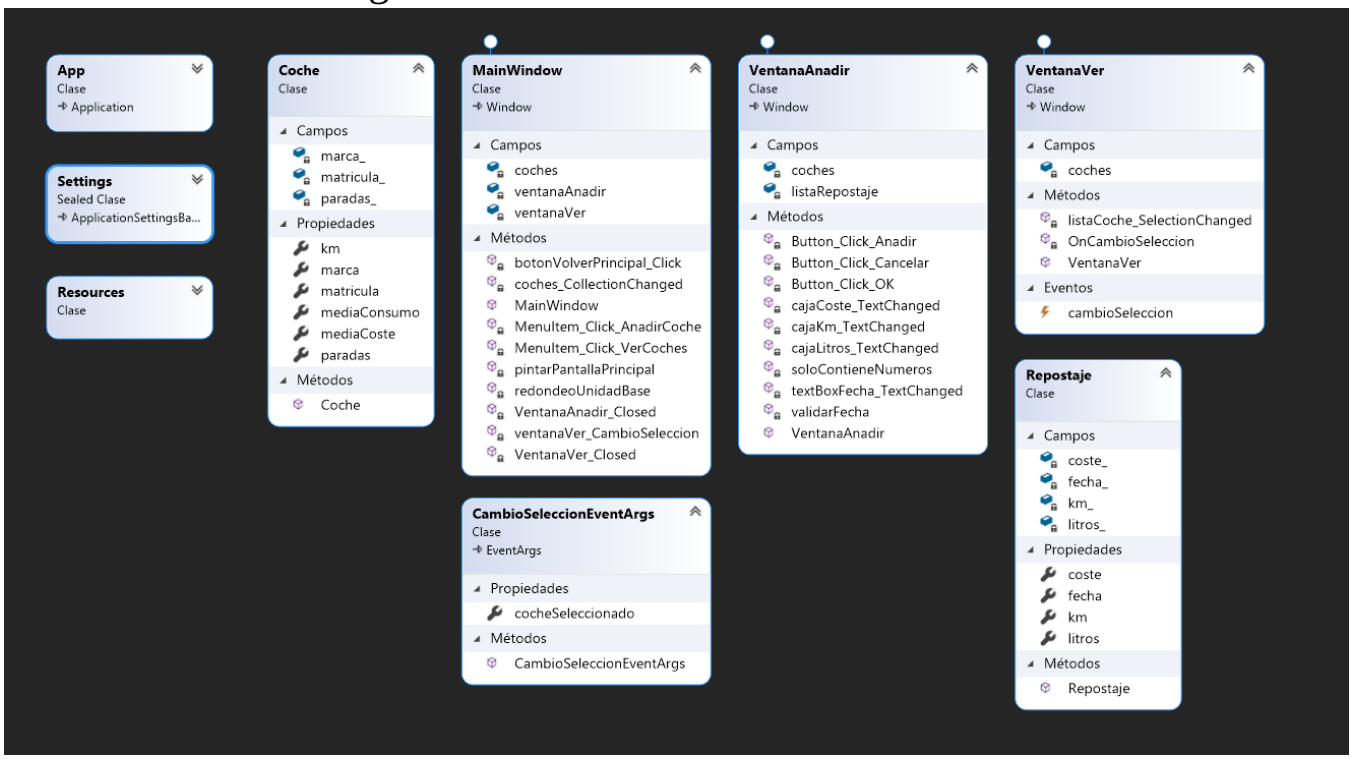
1.2.4.2. Gráficos individuales de coche

En la ventana secundaria de "Ver Coches" al seleccionar un coche en la lista, en la pantalla principal se muestra un gráfico con la evolución de los diferentes repostajes. Aparecen la evolución del gasto, del consumo de litros y de los kilómetros recorridos entre cada repostaje.

2. Manual del programador

En este apartado se hará un breve repaso a nivel técnico del programa. Se expondrá primero un diagrama de las clases utilizadas en el sistema, se explicarán dichas clases, los métodos más relevantes de las mismas y las relaciones especiales entre objetos como pueden darse a través del uso de eventos.

2.1. Diagrama de clases



2.2. Clases

Las clases utilizadas en el programa son las siguientes.

- Repostaje: representa una parada de repostaje de un coche. Los atributos son los siguientes:
  - Coste_: coste de echar gasolina al coche en euros. Es un número entero.
  - Fecha_: fecha en la que se hace el repostaje. Tipo DateTime.
  - Km_: kilómetros en el cuentakilómetros en el momento del repostaje. Tipo entero.
  - Litros_: litros de gasolina. Tipo entero.
- Coche: representa a un coche. Los atributos son los siguientes:
  - Marca_: marca del coche. Tipo string.
  - Matricula_: matricula del coche. Tipo string.
  - Paradas_: lista de objetos "Repostaje" tipo ObservableCollection.

  Las propiedades más relevantes son:

  - Km: devuelve el valor más alto del cuentakilómetros del coche revisando los objetos "Repostaje" de la lista "paradas_".

        - mediaConsumo: devuelve una media del consumo realizado por 100 km recorridos.
        - mediaCoste: devuelve una media del coste por cada 100 km recorridos.
- MainWindow: clase derivada de Window que representa la ventana principal. En ella se plasman los gráficos y se accede a las funciones. Los atributos que tiene son:
    - Coches: colección de objetos "Coche" tipo ObservableCollection que es el modelo de la aplicación, pues tienen acceso a ella todas las ventanas.
    - ventanaAnadir: dado que es MainWindow la ventana principal, crea la ventana secundaria. Esta ventana se usa para añadir coches a la colección.
    - ventanaVer: otra ventana secundaria, se usa para visualizar todos los coches registrados y para poder ver sus repostajes y seleccionar sus gráficas individuales.

    Los principales métodos son los siguientes:

    - cochesCollectionChanged: gestor de evento que se activa cada vez que se añade un elemento a la colección.
    - pintarPantallaPrincipal: dibuja todos los parámetros de todos los coches en el lienzo principal. Es llamado por cochesCollectionChanged.
    - ventanaVerCambioSeleccion: gestiona un evento de cambioSeleccion generado por la ventanaVer al cambiar la selección de la lista de coches. Dibuja en el lienzo el gráfico individual del coche seleccionado.
- VentanaAnadir: ventana secundaria utilizada para añadir coches a la colección. Los atributos que usa son:
    - Coches: lista de coches. Se la pasa MainWindow por referencia al crear la ventana.
    - listaRepostajes: lista de objetos "Repostaje" que se usa para crear el coche.

    Los métodos usados son:

        - buttonClick_Anadir: gestor del evento "Click" del botón botonAnadir. Sirve para añadir objetos repostajes a la lista listaRepostajes que luego se le pasará al coche al crearlo.

        - buttonClick_OK: gestor del evento “Click” del botón “botonOK”. Sirve para añadir el nuevo “Coche” a la colección.
        - buttonClick_Cancelar: gestor del evento “Click” del botón “botonCancelar” sirve para abortar la operación de añadir un coche a la colección.
- VentanaVer: ventana secundaria cuya finalidad es la de visualizar la colección de coches y los datos de los repostajes de cada coche. Los atributos que tiene son:
    - Coches: colección de coches. Ya ha sido explicado anteriormente, se le pasa a esta ventana por parte de MainWindow al crearla.

    Los métodos que usa son los siguientes:

    - listaCoche_SelectionChanged: gestor asociado al evento “SelectionChanged” de la listView principal. Hace dos funciones, muestra en la listView secundaria los datos de los repostajes del coche seleccionado y llama al método OnCambioSeleccion.
    - OnCambioSeleccion: activa el evento “CambioSelección” cuya función es avisar a MainWindow para que dibuje la gráfica individual del nuevo coche seleccionado en el listView.
- CambioSeleccionEventArgs: clase derivada de EventArgs cuya función es la de ir asociada al evento CambioSeleccion e incluir el Coche seleccionado en el listView de la ventana secundaria VentanaVer.
- App, Settings y Resources: clases creadas por WPF.

2.3. Relaciones entre objetos. Eventos

La única relación de objetos entre eventos es la que se ha explicado en el apartado anterior con el evento “CambioSeleccion”. Este evento se activa cuando se cambia la selección en el listView principal de VentanaVer y su principal cometido es el de notificar a MainWindow que dibuje el gráfico individual del Coche nuevo seleccionado, que va incluido en la clase CambioSeleccionEventArgs.

3. Referencias

- PDFs de la asignatura “Interfaces Gráficas de Usuario” de 3º de Ingeniería Informática de la USAL.
- https://learn.microsoft.com/es-es/dotnet/api/system.datetime.compare?view=net-7.0
- https://learn.microsoft.com/es-es/visualstudio/ide/class-designer/how-to-add-class-diagrams-to-projects?view=vs-2022
- https://stackoverflow.com/questions/8844674/how-to-round-to-the-nearest-whole-number-in-c-sharp
- https://stackoverflow.com/questions/7461080/fastest-way-to-check-if-string-contains-only-digits-in-c-sharp
- https://learn.microsoft.com/en-us/dotnet/api/system.string.isnullorwhitespace?view=net-7.0
- https://learn.microsoft.com/en-us/dotnet/api/system.string.equals?view=net-7.0#system-string-equals(system-string-system-stringcomparison)
- https://learn.microsoft.com/es-es/dotnet/api/system.windows.controls.textchangedeventargs?view=windowsdesktop-7.0
- https://stackoverflow.com/questions/34403009/how-to-clear-the-contents-of-a-canvas-in-wpf
- https://learn.microsoft.com/es-es/dotnet/api/system.collections.objectmodel.observablecollection-1?view=net-7.0