

INTERFACES GRÁFICAS-TRABAJO FINAL

MANTENIMIENTO DE AUTOMÓVILES

Manual de Usuario:

Esta aplicación, está enfocada a las personas que quieran monitorizar alguna flota de vehículos.

Mi aplicación tiene 9 ventanas, en las dos primeras, esta aplicación permite registrar una cuenta y asociar una base de datos con esa cuenta y después iniciar sesión, si no se quiere registrar hay una cuenta general (usuario/usuario), que ya tiene en su base de datos unos valores de prueba.

Iniciar Sesión
Usuario

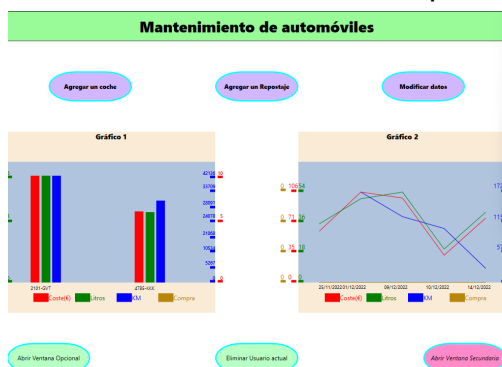
Contraseña

Registrarse
Usuario

Contraseña

Nombre de tu BD

La principal, es la main (a la cual accedemos después de iniciar sesión con una cuenta registrada) y desde ahí podremos ir a la mayoría de las otras ventanas. En la main aparecen dos gráficas. La primera, contendrá un gráfico de barras sobre la media de litros y de coste, los kilómetros totales y la media de precio de los productos comprados en tienda de cada vehículo. La segunda gráfica, se creará cuando se pulse sobre la tabla uno de la ventana secundaria, se creará una gráfica de polilíneas que representará el coste, litros, precio de productos y los kilómetros hechos desde el repostaje anterior por cada fecha.



Además hay 5 botones:

Datos del vehículo:

Matrícula:

Marca:

Fecha:
Seleccione una fecha

Kilometraje:

Litros:

Coste:

Añadir

Al pulsar el botón de agregar vehículo aparecerá una nueva ventana que te dejará introducir la matrícula, marca, fecha, kilómetros, litros y coste. El vehículo introducido modificará la gráfica de barras, y la tabla uno de la ventana secundaria.

Nuevo repostaje:

Elige la matrícula:
2101-GVT
4785-KKK

Fecha:
Seleccione una fecha

Kilometraje:

Litros:

Coste:

Añadir

Al pulsar el botón de agregar repostaje aparecerá otra ventana con una lista de matrículas al seleccionar una podrás introducir fecha, litros, coste y kilómetros de su último repostaje.

Modificar Coche:

2101-GVT
4785-KKK

Modificar Repostajes

Matrícula:
2101-GVT

Marca:
Peugeot

Modificar

Pulsando el botón modificar, se creará otra ventana que nos dejará seleccionar un coche y podremos modificar su matrícula y su marca, esta ventana también tiene un botón que al tener una matrícula seleccionada de la lista, abrirá otra ventana y se podrá modificar su último repostaje.

Los dos botones de la zona baja de la aplicación dan lugar a una ventana cada uno.

El botón abrir la ventana secundaria abrirá una ventana secundaria, si la anterior ventana secundaria estuviera cerrada.

En la ventana secundaria hay dos tablas, la primera representa los datos generales de cada vehículo (como la gráfica de barras) y la segunda tabla representa los datos de cada repostaje (como la gráfica de las polilíneas).

Además hay tres botones en esta ventana, para borrar el vehículo seleccionado, para borrar el repostaje seleccionado y para guardar los datos de las tablas en nuestra base de datos.

Tabla 1					Tabla 2				
Matricula	Marca	Kilometraje	Media de consumo por 100km	Media	Fecha	KM	Litros	Coste	Compra en tienda
2101-GVT	Peugeot	42136	5.617978	10.39	11/24/2022 12:00:00 AM	40000	50	89	0
4785-KKK	Citroen	32326	3.70543	6.921	11/30/2022 12:00:00 AM	40800	30	58	0
					12/7/2022 12:00:00 AM	41500	40	75	0
					12/9/2022 12:00:00 AM	42136	32	61	0

Borrar todo el vehículo

Añadir a BD

Borrar el repostaje


El botón abrir ventana opcional, abrirá una tienda en la que seleccionando un coche y posteriormente un repostaje podremos añadir al carrito y comprar los productos de este. Al comprar los productos las gráficas y tablas de las otras ventanas se ven modificadas y actualizadas.

TIENDA


Matricula:
2101-GVT
4785-KKK

Fecha Repostaje:
11/24/2022 12:00:
11/30/2022 12:00:
12/7/2022 12:00:0
12/9/2022 12:00:0


Producto:



4 €



1,5 €



1,1 €

Carrito:
Anticongelante 4 € 2
Chocolate 1,5 € 1

Añadir al carrito

Borrar del carrito

Comprar

El botón Eliminar usuario actual eliminará la base de datos asociada y el usuario y contraseña que se estén usando en ese momento.

Manual del programador:

Este es mi diagrama de objetos con mis eventos, clases y ventanas



Primero vamos a explicar las clases más pequeñas:

Matricula: pedimos un int y un string y lo juntamos para hacer la matrícula.

Datos: pedimos una matrícula, un string, tres float y una observable collection de repostaje para así luego crear la media de litros, la media de coste y también (aunque es opcional) la media de dinero gastado en la tienda, además tiene dos funciones para convertirse a string y a un string dividido con puntos.

Repostaje: pedimos tres float y un datetime y lo guardamos normalmente en una observable collection para saber a qué matrícula pertenece, también guarda lo gastado en tienda por repostaje. Otras funciones son pasar a string y pasar a un string dividido por puntos.

Productos: Sirve para guardar 3 strings y un int, para posteriormente crear la tienda.

Producto: Sirve para guardar el producto de la tienda seleccionado por el cliente.

Comprado: Guarda el precio de un producto y guarda cuántos productos ha comprado el cliente para calcular el precio total de cada repostaje.

Ahora vamos a explicar las ventanas y sus clases:

Login: Ventana con la que inicia el programa. Hay dos opciones iniciar sesión, que comprueba si los datos introducidos concuerdan en la base de datos y si son correctos abren la MainWindow con la base de datos de ese usuario.

Si el usuario da a registrarse se abrirá una nueva ventana para crear un usuario nuevo.

VRegistrarse: Su evento más importante repasa si el usuario o la base de datos ya está utilizados si es así hace repetir el dato repetido, cuando te registras tienes usuario y base de datos propia.

VRepostaje: Su único método importante es añadir click con el que compramos nuestros datos y el elemento seleccionado de la list view y cuando son iguales, permitimos al usuario agregar un nuevo repostaje, pero antes comparamos que todo los datos estén bien introducidos, posteriormente llamamos a los eventos necesarios para actualizar la gráfica de barras y las tablas por que al introducir un nuevo repostaje ya se actualizan las polilíneas.

VCoche: Su único método importante es añadir click con el que compramos nuestras matrículas y la matrícula nueva y si no está en nuestra ObservableCollection añadimos un nuevo coche, pero si concuerda con una nueva permitimos al usuario agregar un nuevo repostaje, pero antes comprobamos que todo los datos estén bien introducidos.

VModificar: Aparte de crear la ventana VModificarR, su función más importante es el método modificar clic permite al usuario modificar la matrícula o marca del coche seleccionado en la lista view. Y posteriormente llama a los eventos necesarios para actualizar las tablas y las gráficas.

VModificarR: El método importante es Modificar click que permite, al seleccionar el repostaje de la listview te permite modificar los litros, kilómetros y coste del repostaje. Posteriormente se actualizan las tablas y las gráficas en la ventana VModificarR.

Ventana: Tiene dos listview, el evento selectionChanged de la primera listview, limpia la segunda listview por si tenía elementos posteriores y muestra la segunda listview completado por los repostaje del coche seleccionado, además hay dos botones que sirven para borrar vehículos y repostajes, respectivamente. Además hay otro método que rellena un txt con los datos de cada vehículo que tengamos en las observablecollection.

Opcional: Esta ventana tiene 4 listview una para mostrar los coches, otra para mostrar los repostajes de cada coche seleccionado, otra para enseñar los productos y los precios de cada artículo y por último, una al principio vacía que cuando el usuario va dando a añadir al carrito los productos van llenando esta listview y cuando va dando a borrar los artículos desaparecen.

El método más importante es el evento que crea el botón comprar, este busca cual es el repostaje seleccionado y una vez lo encuentra, asigna precio total de la compra a ese repostaje y limpia el carrito para la siguiente compra. También gestiona los eventos para actualizar las tablas y las gráficas.

MainWindow: Esta ventana aparte de los botones para crear las otras ventanas y los eventos para no abrir la misma ventana dos veces a la vez tiene dos métodos muy importantes crearBarras y crearPolilineas.

CrearBarras lo que realiza es que por vehículo dato va a crear cuatro barras una para los kilómetros totales, otra para la media de litros, otra para la media de coste y otra para la media de lo gastado en tienda. Estos rectángulos los creamos cogiendo referencia del ancho y del alto del canvas y así podemos usar un evento para que

cuando el tamaño del canvas cambie el tamaño de nuestros rectángulos también vayan cambiando. En mi caso yo me he ayudado de una regla de tres para calcular las proporciones. Lo que hago es coger el valor mayor de cada barra y asignarle a esa barra el mayor valor del canvas y a las demás les he dado el valor proporcional. Una vez hechos los cuadrados mi programa realiza los ejes en los que crea textblocks para enseñar las matrículas en el eje x y en el eje y los valores de cada barra, el valor máximo será el valor de la barra más alta por cada tipo.

El método crear polilíneas trabaja parecido la única diferencia es que en vez de con cada vehículo lo hacemos por cada repostaje del vehículo seleccionado, además cambiamos los rectángulos por puntos crearemos por cada fecha cuatro puntos (litros, costes, kilómetros desde el último repostaje y precio de lo comprado en tienda). La creación de estos puntos es igual que en la de barras, utilizo una regla de tres. después cada punto lo meto en su respectiva colección de puntos y después cuando se acaban los repostajes, meto a las polilíneas su colección de puntos. Los ejes son iguales que los de barras, simplemente cambia que en el eje x en vez de matrícula son fechas y en el eje y no son la media de litros son los litros, etc....

Por último, me queda explicar los eventos, esta aplicación tiene un total de 4 EventArgs creados por mí.

crearBarrasEventArgs: Este EventArgs lo uso en la mayoría de ventanas, para llamar al evento que se lanza a la main, en esta llama al método crearBarras y así actualizamos la gráfica de barras.

crearpolilineasEventArgs: Este EventArgs lo uso en la mayoría de ventanas, para llamar al evento se lanza a la main, en esta llama al método crearpolilineas y así actualizamos la gráfica de barras.

actualizartablasEventArgs: Este EventArgs lo uso en la mayoría de ventanas, para llamar al evento que se lanza desde la main hasta la ventana Ventana y ahí actualizar las listView.

VMNotificarEventArgs: uso a este EventArgs para llamar a un evento y que la ventana VModificarR pueda decirle a la ventana VModificar que ya ha acabado y así poder actualizar las tablas y las gráficas desde esa ventana.

Bibliografía:

- Creación de gráficos
<https://learn.microsoft.com/es-es/archive/msdn-magazine/2009/september/charting-with-datatemplates>
- Listbox
https://www.youtube.com/watch?v=5BXzIzb_0uQ
- Ficheros
<https://www.youtube.com/watch?v=o6rvnqv40WI&list=PL8gxzfBmzgexdFa0XZZSZZn2Ogx3j-Qd5&index=46>

<https://www.youtube.com/watch?v=P8G7LAd3dMA&list=PL8gxzfBmzgexdFa0XZZSZZn2Ogx3j-Qd5&index=44>
<https://easynetstudio.wixsite.com/easynetstudio/crear-y-modificar-txt-en-c-sharp>

- Polilíneas
<https://learn.microsoft.com/en-us/dotnet/api/system.windows.forms.control.sizechanged?view=windowsdesktop-7.0>
- Cancel Event Handler
<https://learn.microsoft.com/en-us/dotnet/api/system.componentmodel.canceleventargs.cancel?view=net-7.0>
- Opcional
<https://www.tutorialesprogramacionya.com/csharpya/puw/detalleconcepto.php?punto=17&codigo=17&inicio=0>

Conclusión

Esta asignatura me ha resultado muy interesante tanto su contenido como su forma de evaluar, ojalá fueran todas así, he aprendido mucho sobre c# y los lenguajes orientados objetos, eso sí, no todo ha sido bueno, igual que he dicho que me ha gustado la idea de que un trabajo sea la nota final de la asignatura, también me he estresado mucho realizando la práctica pensando si algo tenía acoplamiento por hacerlo de alguna forma. Pero en general si por mi fuera haría todas las asignaturas así, aunque quitase mucho tiempo.