



## **Informe del Proyecto 1 - Parte 1: Automóviles**

Fabricio Josue Lopez Reyes, 202210190

Isabella Luana Romero Celis, 202210302

Gabriel Eduardo Romero Diez, 202210303

Valeria Celeste Valdez Campoverde, 202210359

Kharol Emilia Villamonte Montesinos, 202210373

Josue Yip Rodriguez, 202120039

Ciencias de la Computación, Universidad de Tecnología

CS1111 - Programación 1

Jesus Edwin Bellido Angulo // Carlos Alberto Sotomayor Beltran

2022

**1. Resumen:**

El programa cuenta un sistema de elección de autos para mejorar la experiencia del usuario, además de una base de datos donde se registra cada uno de ellos la venta de estos y la modificación de ellos en la base de da

## 2. Introducción:

Durante los últimos años, la industria automotriz en el Perú no ha hecho más que crecer. Hasta abril del 2022, la información más reciente de la Superintendencia Nacional de Registros Públicos (Sunarp) revelaba que en el segmento de cuatro meses se vendieron 52 904 unidades de vehículos livianos, un incremento del 11,6% comparado al 2021. De este incremento, se estima que 12,913 unidades son automóviles (Gerencia de Estudios Económicos de la AAP, 2022).

Por otro lado, de acuerdo con el Índice Nacional del “Flujo Vehicular por Unidades de Peaje” (INEI, 2022) que registra movimientos de entradas y salidas del tráfico vehicular creció un 4,5% en comparación al 2021. De este porcentaje, un 5,7% eran vehículos ligeros. De la misma manera, la venta e inmatriculación de los mismos ha incrementado en más de 10 000 unidades durante el periodo Enero - Mayo.

De acuerdo a las estadísticas presentadas, se puede inferir la importancia de la industria automotriz en el Perú, así como de su adecuado registro, pues las estadísticas son de utilidad al momento de plantear políticas de transporte o como un indicador de actividad y desarrollo económico en diversas regiones según su tráfico vehicular. De esta manera, se considera que el presente trabajo sería de gran ayuda a este fin.

### 3. Definiciones:

#### 3.1. *menu\_opcion()*:

Esta función se encarga de mostrar cuatro opciones numeradas (registrar auto, autos disponibles, compra auto y salir) en un formato de menú. Además permite al usuario seleccionar qué opción quiere implementar a continuación.

#### 3.2. *separar\_por()*:

Esta función separa los datos del archivo “registro.txt”, donde se almacenan los datos de los carros, para que puedan ser legibles para el programa.

#### 3.3. *registro\_auto()*:

Esta función se encarga de almacenar características (marca, año de fabricación, color y precio) ingresadas por el usuario, para luego ser mostradas en la relación de autos disponibles en un formato de tabla. Además mediante el uso de la librería os corrobora la existencia del *archivo.txt*, de lo contrario, procede a crearlo.

#### 3.4. *inventario\_auto()*:

Esta función se encarga de mostrar la relación de los autos disponibles para la venta según el inventario de la tienda.

#### 3.5. *filtrar\_por()*:

Esta función se encarga de filtrar el inventario de auto para luego mostrarnos una matriz (almacenamiento de información ordenada) de todos los autos que coincidan con las características previamente ingresadas en la sección de registro de auto, hasta que el usuario compre el auto que desee.

#### 3.6. *comprar\_auto()*:

Esta función permite al usuario comprar un auto que se encuentre disponible al introducir datos como (marca, año de fabricación, color y precio) para posteriormente mostrar que el carro ha sido vendido satisfactoriamente.

### 3.7. *reemplazar\_vendido()*:

En el caso de que un carro sea vendido, esta función se encarga de reemplazar el estado del carro (“disponible” por “vendido”).

### 3.8. *eliminar\_vendido()*:

Detecta los carros vendidos para que sean eliminados y muestre los nuevos carros disponibles con un nuevo id asignado acorde al total de autos.

### 3.9. *si\_queda\_uno()*:

Busca que la opción de compra de auto sea sea válida para luego ejecutar las funciones *reemplazar\_vendido* y *eliminar\_vendido*.

### → *librería tabulate*:

Esta librería permite al usuario imprimir o escribir un texto en tablas en distintos formatos.

### → *librería os*:

Esta librería permite interactuar con el sistema operativo de la computadora, desde crear, mover y eliminar archivos.

#### 4. Funcionamiento del programa:

- **Función `menu_opcion()`:**

La función `menu_opcion`, abre el `menu.txt` y lo imprime en pantalla.

El cual muestra las siguientes opciones:

1. Registrar Auto
2. Autos Disponibles
3. Compra Auto
4. Salir

```
#Función para mostrar el menú de opciones
def menu_opcion():
    with open("menu.txt", "r") as abrirMenu:
        print(abrirMenu.read())
    while True:
        opcion = input("opcion: ")
        #if opcion < 4 and opcion > 0:
        if opcion == "1":
            registro_auto()
            enterContinuar()
            break
        elif opcion == "2":
            inventario_auto()
            enterContinuar()
            break
        elif opcion == "3":
            comprar_auto()
            enterContinuar()
            break
        elif opcion == "4":
            exit()
        else:
            print("Introduce una opcion valida, por favor vuelva a intentarlo")
```

Si se elige la opción “**Registrar Auto**”, se ejecuta las funciones “*registro\_auto*” y “*enterContinuar*”.

- **Función *registro\_auto()*:**

Abre un menú que permite registrar un auto por las siguientes características: marca, año de fabricación, color y precio.

Si es que los datos ingresados son erróneos, se le pide al usuario ingresar nuevamente.

```
#Registro de autos
def registro_auto():
    try:
        marca = input("Introduce la Marca: ")
        fabricacion = int(input("Introduce el año de Fabricación: "))
        color = input("Introduce el Color: ")
        precio = int(input("Introduce el Precio: "))
    except:
        print("-----")
        print("Ingresaste un valor erroneo, por favor vuelve a ingresar los datos: ")
        print("-----")
        registro_auto()
    disponibilidad = "Disponible"
    if os.path.exists("registro.txt"):
        with open("registro.txt", "r") as file:
            contador = 1
            for line in file:
                contador += 1
            registro = "{},{},{},{},{},{}\n".format(contador,marca.lower().capitalize() , fabricacion,
            color.lower().capitalize() , precio, disponibilidad.lower().capitalize())
            with open("registro.txt", "a") as file:
                file.write(registro)
    else:
        contador = 1
        registro = "{},{},{},{},{},{}\n".format(contador,marca.lower().capitalize() , fabricacion,
        color.lower().capitalize() , precio, disponibilidad.lower().capitalize())
        with open("registro.txt", "a") as file:
            file.write(registro)
    #Fin del programa e interacción con el usuario
    print("\n Registro exitoso !!!")
```

- **Funcion *enterContinuar()*:**

Detecta si el usuario presionó la tecla “ENTER” para luego llevarlo a la función *menu\_opcion*.

```
def enterContinuar():
    a = input("Presiona enter para continuar...")
    a = ""
    if a == "":
        menu_opcion()
```

Una vez registrado el auto, el programa llama a la función *enterContinuar*, acto siguiente llama a la función *menu\_opcion* para seguir con otras operaciones.

Si se elige la opción “Autos Disponibles”, se ejecuta las funciones “*inventario\_auto*” y “*enterContinuar*”.

- ***Función inventario\_auto():***

Abre el archivo *registro.txt* en donde se tienen registrados todos los autos y todas sus características.

Acto siguiente los muestra en pantalla en forma de una tabla donde se muestran los autos registrados y disponibles. Cada uno con sus diferentes características; marca, fabricación, color y precio.

```
#Mostrar el inventario de la tienda(los autos disponibles)
def inventario_auto():
    with open("registro.txt", "r") as lineas:
        listaLineas = []
        listaLineas = (lineas.readlines())
        matrisita = []
        for i in listaLineas:
            lineax = i.split(",")
            matrisita.append(lineax)

    #Imprimo el título
    with open("titulo_autos_disponibles.txt", "r") as file:
        print(file.read())

    #Imprimo la tablita
    print(tabulate(matrisita, headers = ["" , "Marca", "Fabricacion", "Color",
    "Precio", "Estado"], tablefmt="fancy_grid" ))
```

- ***Funcion enterContinuar():***

Detecta si el usuario presionó la tecla “ENTER” para luego llevarlo a la función *menu\_opcion*.

```
def enterContinuar():
    a = input("Presiona enter para continuar...")
    a = ""
    if a == "":
        menu_opcion()
```



Una vez se muestran los autos disponibles, el programa llama a la función *enterContinuar*, acto siguiente llama a la función *menu\_opcion* para seguir con otras operaciones.

Si se elige la opción “**Comprar Auto**”, se ejecuta las funciones “*comprar\_auto*” y “*enterContinuar*”.

- **Función *comprar\_auto()*:**

Abre el archivo *compra\_de\_autos.txt* que muestra e imprime el título.

Esta función llama a una función “*filtrar\_por*”, “*si\_queda\_uno*”, “*reemplazar\_vendido*”, “*eliminar\_vendido*”.

```
def comprar_auto():
    #Mostrar título de compra de autos
    with open("compra_de_autos.txt", "r") as abrirCompra_Autos:
        print(abrirCompra_Autos.read())

    #Creación de la nueva matriz que se encargará de los otros datos
    with open("registro.txt", "r") as lineas:
        listaLineas = []
        listaLineas = (lineas.readlines())
        matrisita = []
        for i in listaLineas:
            lineax = i.split(",")
            matrisita.append(lineax)
```

- **Función *filtrar\_por()*:**

Filtra a pedido del usuario los autos por las características de estos; marca, año de fabricación, color o precio.

Una vez filtrados los agrupa en una nueva matriz la cual se muestra al usuario.

```
#Filtro de selección para comprar auto
def filtrar_por(dato, matriz_ingresada, posicion):
    nueva_matriz = []

    #Filtrar por dato
    for i in range(0, len(matriz_ingresada)):
        if matriz_ingresada[i][posicion] == dato :
            nueva_matriz.append(matriz_ingresada[i])

    #imprimo titulo
    with open("titulo_autos_disponibles.txt", "r") as file:
        print(file.read())

    #Devuelvo la matriz de los autos seleccionados por el usuario
    return nueva_matriz
```

- ***Función si\_queda\_uno():***

Cuando el usuario termina de escoger el auto y sólo queda uno en el registro filtrado, le pregunta al usuario si desea comprar el auto que sobra.

```
def si_queda_uno(auto_a_comprar):
    if len(auto_a_comprar) == 1:
        strauto_a_comprar = ",".join(auto_a_comprar[0])
        strauto_a_comprar = strauto_a_comprar + "\n"

        print("Deseas comprar este auto?: ")
        comprar = input("1.- sí\n2.- Volver al menú\n3.-Salir\nTu respuesta: ")
        if comprar == "1":
            #Codigo reutilizado de "inventario_auto()"
            reemplazar_vendido("registro.txt", int(auto_a_comprar[0][0]))
            print("Gracias por su compra")
            inventario_auto()
            eliminar_vendido("registro.txt")
            enterContinuar()
        elif comprar == "2":
            menu_opcion()
        elif comprar == "3":
            exit()
        else:
            print("Introduce una opción válida")
            print("Deseas comprar este auto?: ")
            comprar = input("1.- sí\n2.- volver\n3.-Salir\nTu respuesta: ")
    elif len(auto_a_comprar) == 0:
        print("Lo sentimos no disponemos el auto que deseas comprar")
        print("Vuelve a intentarlo")
        enterContinuar()
```

- ***Función reemplazar\_vendido():***

Filtra y busca el auto que compró y cambia su estado a “Vendido”.

```
def reemplazar_vendido(file, x):
    temporal = []
    with open(file, 'r') as f:
        contador = 1
        for line in f:
            if contador == x:
                a = line.split(",")
                a[-1] = "Vendido\n"
                b = ",".join(a)
                temporal.append(b)
                contador += 1
            else:
                temporal.append(line)
                contador += 1
    os.remove(file)
    with open(file, 'w') as f:
        for line in temporal:
            f.write(line)
```

- ***Función eliminar\_vendido():***

Filtra y busca todos los autos con el estado “**Vendido**” y los elimina del registro.

```
def eliminar_vendido(file):
    temporal = []
    with open(file, 'r') as f:
        contador = 1
        for line in f:
            a = line.split(",")
            if a[-1] == "Vendido\n":
                pass
            else:
                n = a[1:-1]
                b = ",".join(n)
                z = "{},{},Disponible\n".format(contador, b)
                temporal.append(z)
                contador += 1
    os.remove(file)
    with open(file, 'w') as f:
        for line in temporal:
            f.write(line)
```

- ***Funcion enterContinuar():***

Detecta si el usuario presionó la tecla “ENTER” para luego llevarlo a la función *menu\_opcion*.

```
def enterContinuar():
    a = input("Presiona enter para continuar...")
    a = ""
    if a == "":
        menu_opcion()
```

Una vez se muestran los autos filtrados, el programa llama a la función *enterContinuar*, acto siguiente llama a la función *menu\_opcion* para seguir con otras operaciones.

Si se elige la opción “**Salir**”, se cierra el programa.

En todo caso se reciba una opción que no esté en las brindadas, se le pide al usuario ingresar una de las opciones dadas.

## 5. Ejemplos de ejecución:

## 6. Conclusiones y recomendaciones:

- El software diseñado es ideal para una compañía tipo MYPE (micro y pequeñas empresas) que se dedique al rubro de la compra y venta de automóviles. Puesto que el sistema cuenta con una interfaz minimalista y efectiva. Se recomienda al usuario no borrar ni modificar el formato predeterminado del *archivo.txt*, ya que es donde todos los datos se encuentran almacenados.
- Con el uso de las librerías **pandas**, **csv**, **pyautogui** podríamos hacer que el código funcione de una manera más eficiente. Se recomienda elaborar una función que filtre o vea de una manera de contar y registrar si se tiene la existencia de un auto con las misma características.
- Se podría utilizar una base de datos actualizada o una base de datos en la nube para evitar pérdidas locales y optimizar la velocidad de ejecución del software.

## 7. Fuentes:

- Gerencia de Estudios Económicos de la AAP. (2022, mayo). INFORME DEL SECTOR AUTOMOTOR ABRIL 2022.  
<https://aap.org.pe/informes-estadisticos/abril-2022/Informe-Abril-2022.pdf>
- Anónimo. (2022). Flujo vehicular por unidades de peaje. 12/06/22, de INEI Sitio web:  
[https://www.inei.gob.pe/media/MenuRecursivo/boletines/informe-flujo-vehicular\\_1.pdf](https://www.inei.gob.pe/media/MenuRecursivo/boletines/informe-flujo-vehicular_1.pdf)
- Anónimo. (2022). Venta e Inmatriculación de Vehículos Nuevos 2022. 12/06/22, de AAP Sitio web:

[https://aap.org.pe/estadisticas/ventas\\_inmatriculaciones\\_vehiculos\\_nuevos/venta-e-inmatriculacion-de-vehiculos-nuevos-2022/](https://aap.org.pe/estadisticas/ventas_inmatriculaciones_vehiculos_nuevos/venta-e-inmatriculacion-de-vehiculos-nuevos-2022/)