

Proyecto
Introducción A La Programación
I Cuatrimestre

Camila Argeñal Coronado
Eduardo miranda mora
Jordan Ivan Soto Morales
Josias Aguilar Arroyo
Samantha Perez Rojas

Universidad Fidelitas
Sede San Pedro

Prof. José Antonio Ortega Gonzalez

6 de Marzo del 2024

Código acceso a Git hub Repositorio.

<https://github.com/ivansotomorales31/python-project-4.git>

Avance I. 8 de marzo del 2024

Código

```
inventario_vehiculos = []
Registro_Cedulas = []

def agregar_vehiculo(marca, año, modelo, cilindraje, precio_alquiler, precio_vehiculo,
placa, cantidad):
    if placa not in inventario_vehiculos:
        inventario_vehiculos[placa] = {'marca': marca,
                                        'año': año,
                                        'modelo': modelo,
                                        'cilindraje': cilindraje,
                                        'precio_alquiler': precio_alquiler,
                                        'precio_vehiculo': precio_vehiculo,
                                        'cantidad_disponible': cantidad,
                                        'habilitado': True}
        print("Vehiculo agregado exitosamente")

def reservar_vehiculo(placa, cantidad):
    if placa in inventario_vehiculos:
        if inventario_vehiculos[placa]['habilitado']:
            if inventario_vehiculos[placa]['cantidad_disponible'] >= cantidad:
                inventario_vehiculos[placa]['cantidad_disponible'] -= cantidad
                print("Reserva realizada correctamente.")
            else:
                print("No hay suficientes vehiculos ")
        else:
            print("El vehiculo esta inhabilitado.")
    else:
        print("No existe ningun vehiculo con esa placa.")

def inhabilitar_vehiculo(placa):
    if placa in inventario_vehiculos:
        inventario_vehiculos[placa]['habilitado'] = False
        print("Vehiculo inhabilitado correctamente.")
    else:
        print("No se encontro ningun vehiculo con esa placa.")

def menu_administrador():
    opc = ""
    while opc != "3":
        print()
        print("-----Menu-----")
        print("[1] Gestion inventario Vehiculos")
        print("[2] Gestion de clientes")
        print("[3] Visualizar vehiculos ")
```

```

opc = input("Seleccione una opcion: ")

if opc == "1":
    print("----Gestion inventario vehiculos----")
    print("[1] Agregar vehiculos")
    print("[2] Inhabilitar vehiculos")
    opc_inventario = input("Seleccione una opcion: ")

    if opc_inventario == "1":
        print()
        print("Agrega el vehiculo que deseas")
        Marca = input("Ingrese la marca del vehiculo: ")
        Año = input("Ingrese el año del vehiculo: ")
        Modelo = input("Ingrese el modelo del vehiculo: ")
        cilindraje = input("Ingrese el cilindraje del modelo: ")
        Precio_alquiler = float(input("Ingrese el precio de alquiler: "))
        Precio_auto = float(input("Ingrese el precio del auto: "))
        placa = input("Ingrese el numero de placa")
        cantidad = float(input("Ingrese la cantidad de vehiculos disponibles"))
        agregar_vehiculo(Marca, Año, Modelo, cilindraje, Precio_alquiler, Precio_auto,
placa, cantidad)
    elif opc_inventario == "2":
        placa = input("Ingrese la placa del vehiculo a inhabilitar: ")
        inhabilitar_vehiculo(placa)

elif opc == "2":
    print("Gestion de clientes")
    print("----Menu-----")
    print("[1] Ingresar como invitado")
    print("[2] Ingresar como cliente registrado")
    cli_opc = input("Seleccione una opcion para la gestion de clientes: ")

    if cli_opc == "1":
        print("Cliente invitado")

    elif cli_opc == "2":
        print("Iniciar sesion")
        Nombre = input("Ingrese su nombre:")
        NumCedula = input("Ingrese su numero de cedula:")

        if NumCedula in Registro_Cedulas:
            print("Bienvenido de nuevo",Nombre)

        else:
            print("Cedula no encontrada, porfavor registrese")
            Registro_Cedulas.append(NumCedula)
            Nombre = input("Ingrese su nombre:")
            Telefono = input("Ingrese su numero de telefono:")
            print("Se registro correctamente ")

elif opc == "3":
    print("Visualizar vehículos")
    for i in inventario_vehiculos:
        print(i)
    print("Estos son los vehiculos")

```

```
menu_administrador()
```

Requerimientos técnicos

1. Gestión inventario vehículos

1.1 Diseño

```
inventario_vehiculos = []
Registro_Cedulas = []

def agregar_vehiculo(marca, año, modelo, cilindraje, precio_alquiler, precio_vehiculo,
placa, cantidad):
    if placa not in inventario_vehiculos:
        inventario_vehiculos[placa] = {'marca': marca,
                                        'año': año,
                                        'modelo': modelo,
                                        'cilindraje': cilindraje,
                                        'precio_alquiler': precio_alquiler,
                                        'precio_vehiculo': precio_vehiculo,
                                        'cantidad_disponible': cantidad,
                                        'habilitado': True}
        print("Vehiculo agregado exitosamente")

def reservar_vehiculo(placa, cantidad):
    if placa in inventario_vehiculos:
        if inventario_vehiculos[placa]['habilitado']:
            if inventario_vehiculos[placa]['cantidad_disponible'] >= cantidad:
                inventario_vehiculos[placa]['cantidad_disponible'] -= cantidad
                print("Reserva realizada correctamente.")
            else:
                print("No hay suficientes vehiculos ")
        else:
            print("El vehiculo esta inhabilitado.")
    else:
        print("No existe ningun vehiculo con esa placa.")

def inhabilitar_vehiculo(placa):
    if placa in inventario_vehiculos:
        inventario_vehiculos[placa]['habilitado'] = False
        print("Vehiculo inhabilitado correctamente.")
    else:
        print("No se encontro ningun vehiculo con esa placa.")

def menu_administrador():
    opc = ""
    while opc != "3":
        print()
        print("-----Menu-----")
        print("[1] Gestion inventario Vehiculos")
        print("[2] Gestion de clientes")
```

```

print("[3] Visualizar vehiculos ")
opc = input("Seleccione una opcion: ")

if opc == "1":
    print("----Gestion inventario vehiculos----")
    print("[1] Agregar vehiculos")
    print("[2] Inhabilitar vehiculos")
    opc_inventario = input("Seleccione una opcion: ")

    if opc_inventario == "1":
        print()
        print("Agrega el vehiculo que deseas")
        Marca = input("Ingrese la marca del vehiculo: ")
        Año = input("Ingrese el año del vehiculo: ")
        Modelo = input("Ingrese el modelo del vehiculo: ")
        cilindraje = input("Ingrese el cilindraje del modelo: ")
        Precio_alquiler = float(input("Ingrese el precio de al alquiler: "))
        Precio_auto = float(input("Ingrese el precio del auto: "))
        placa = input("Ingrese el numero de placa")
        cantidad = float(input("Ingrese la cantidad de vehiculos disponibles"))
        agregar_vehiculo(Marca, Año, Modelo, cilindraje, Precio_alquiler, Precio_auto,
placa, cantidad)
    elif opc_inventario == "2":
        placa = input("Ingrese la placa del vehiculo a inhabilitar: ")
        inhabilitar_vehiculo(placa)

```

1.1.1 Codigo usados

1.1.1.1 Funcion Def

<pre>def agregar_vehiculo(ma rca, año, modelo, cilindraje, precio_alquiler, precio_vehiculo, placa, cantidad):</pre>	<pre>Def nombre_funcion(paramet ros): contenido</pre>
--	---

1.1.1.1.1 Forma de aplicación y base (se uso en momentos determinados del codigo)

Este en términos explicativos es un código cuyo propósito es como su diminutivo lo indica crear una nueva función, donde dentro de esta se van a ejecutar una serie de instrucciones, esto se suele utilizar para darle mas automatización a los códigos, creando caminos o “funciones automáticas” para entregar un resultado proveniente de la información que se le brinde a esa función , se sabrá que se esta presenciando una función ya que aunque se les bauticé con un nombre para su identificación siempre se crean con la palabra reservada “**def**”. en las condiciones para usarla están: brindarles parámetros que corresponderán a la informacion-variables que van a manipular dependiendo de la acción que se quiera realizar. Dentro de las variables podemos decir que se pueden usar condicionales, procedimientos, mostrar y agregar etc.

En este código se introdujo de forma externa del código antes del área de código correspondiente al menú principal, mas eso no significa que se vaya a mostrar. El código, permanece inmóvil a menos que sea llamado por decisión del usuario, recibe los datos, en las veces que se uso para el propósito de “gestionar vehículos” se les brindo parámetros de condicionales donde se insertaría el elementos por el cual se quiere guiar, de ese punto definirá sus acciones de guardado de información en arreglos(del cual se hablara después) mostrar información y eliminar información, también a partir de que sus condicionales lo permitan.

1.1.1.1.2 Propósito

Su propósito en este código fue variado, se uso la definición de funciones para guardar información de los vehículos los cuales luego serian ingresados a una lista, su propósito principal también fue darle al código mas automatización, legibilidad y practicidad.

Un ejemplo de su propósito fue el “agregar vehículos”: se llama a esta función desde el código principal, si la placa no se encontraba anteriormente registrada se pasa a una serie de agregados de variables que luego serán incorporados en una lista la cual almacena los conjuntos de informaciones también se guía por medio de la variable “placa” la cual es un gran determinante en muchas funciones y procedimientos dentro del código principal a lo largo de este.

En la función definida como “reserva de vehículos”: se procede a una serie de condiciones que le mostrara al usuario si su vehículo que quiere reservar estaba previamente disponible, en el caso de que no fuera asi se mostraría un mensaje diciéndole que su “vehículo no se encontraba disponible” , entre otras opciones dependiendo del estado del vehículo que fue buscado por los parámetros de “placa” y cantidad

Se utiliza en inhabilitar vehículos donde después de ser la función llamada procede a tomar una serie de condiciones que van a buscar su parámetro y definirán si se encuentra o no, definiendo de esta forma la “deshabilitarían” (se dará una explicación después)

En resumidas cuentas, el propósito de este código acerca de “definir funciones” se encargara de automatizar el código, brindando funciones que si son llamadas significan que deben cumplir con un camino de acciones y que cabe recalcar devolverán al usuario un resultado en este caso también que afectaran códigos como listas externas para cumplir con el propósito previamente definido (afectando las variables en el proceso manipulándolas, todo dependiendo de la acción a elegir) .

Su uso fue de agregar vehículo, reservar vehículos y inhabilitar vehículos dependiendo de los que el usuario pida, será llamada la función que empezara a ejecutarse para después devolver un resultado.

1.1.1.2 condicional if

if placa not in inventario_vehiculos:	if elemento in conjunto_elementos:
if placa in inventario_vehiculos:	

1.1.1.2.1 Forma de aplicación y base

La forma de un if se limita a darle una variable y una condición o serie de condiciones especificas que debe cumplir.

En este código se aplica de forma que de la revisión de una variable en un conjunto de datos en una lista, el código procede (dependiendo de la función que se llame y dependiendo de si la variable ya se encontraba) procede a lanzar datos respectivos a la variable que se dio en la condicional.

En este código podemos caracterizar que usa en una ocasión la diferencia de “if ____ not in ____:” lo cual se refiere a que si la variable no se encontraba en tal conjunto de datos procede a ejecutar acciones, al contrario del código “if ____ in ____:” que verifica que la variable ya sea existente para poder proseguir.

También se agrega el decir que esta codoco se utilizo aparte de las funciones, en un sub menú, mostrado después del menú principal, en opción de “gestión de inventario” seguidamente con los propósitos de que acciones ejecutar según los escrito por el usuario en los “inputs”.

1.1.1.2.2 Propósito

Los propósitos que se le dieron en este código fueron de verificar la existencia de vehículos, en las funciones y después devolver los datos si las variables cumplían con la condición de existir o no dentro de un conjunto de una lista.

Ejemplo de lo dicho antes es su forma de ser aplicada en la función de agregar vehiculos, antes de que se agregue un vehículo, se pasa por la condicional “if ____ not in ____” esto definirá que la variable ya existía antes, para negar el agregado, todo determinado por una variable específica, que seria la variable principal, única y determinante “placa”.

Agregamos que esta variable fue la determinando también en ambas condicionales que se usaron en las funciones de reserva, agregado y inhabilitar.

en la función de inhabilitar se verifica que la variable existiera previamente en la lista respectiva donde se buscaría y luego procedería a eliminarla, en el caso que no, se mandaría un mensaje informando que no existía tal variable en primer lugar.

Un caso particular y específico fue en la función “reserva de vehículo” en la cual el elementos que se quería reservar debe cumplir con varias condiciones creando un subtipo de “if anidado”, dependiendo de la condición que cumpla que mandaría el mensaje de disposición.

Resumiendo su propósito en los menús es dar opciones múltiples y sub-menus para el usuario que el elegirá y que luego dependiendo de estas decisiones, como se dijo anteriormente se llamaran a las funciones.

1.1.1.3 False

Dato= false

1.1.1.3.1 Forma de aplicación y base

El false es un tipo de dato, su función se limita a decir si un dato es como la palabra lo dice “verdadero o falso”

Se aplica en el código dentro de la variable “def inhabilitar_vehiculo(placa):”, donde después de cumplir con condicionales específicas que prueben la existencia de la variable y ser buscada dentro de las listas, se procedería a convertir el dato a falso incluyendo toda la demás información en conjunto que lo acompañan.

1.1.1.3.2 Propósito

Su propósito fue convertir los elementos “vehículos” a falsos, esto automáticamente deshabilitaría los vehículos para no ser opcionales en otras ejecuciones.

1.1.1.4 Llamado de funciones (usado varias veces en el código)

agregar_vehiculo(Marca, Año, Modelo, cilindraje, Precio_alquiler, Precio_auto, placa, cantidad)	Función(parametro1, parametro2)
---	---------------------------------

1.1.1.4.1 Forma de aplicación y base

El llamado de función se aplica colocando el nombre de la función previamente hecha, luego dándole los datos con los que se desea trabajar, luego los datos viajan afuera del código y regresan información ya sean mensajes o simplemente realizan cambios fuera de la vista general del código

1.1.1.4.2 Propósito

Su propósito fue llamar a las funciones, se llamaría a una función dependiendo de lo que el usuario haya seleccionado en el sub menú destinado a la opción 1 en el menú principal (se hablara de este tema después) cada llamado de función tienen la misma finalidad; llamar a la función fuera del código principal, brindarle a la función los datos que necesita para trabajar y luego de que realizara su trabajo se devolviera al menú principal a devolver algún dato o mensaje y haber realizado cambios en los códigos como listas.

En resumidas cuentas su propósito central es llamar funciones para realizar un trabajo.

1.2 Planeamiento

1.2.1 identificación del problema

los problemas que se ven a lo largo del código son los siguientes

- Definir acciones dependiendo de la solicitud del usuario
- Condiciones que se evaluarán si se puede proseguir con lo que el usuario desea
- Existencia de los elementos que se desean buscar, inhabilitar o reservar

1.2.2 acciones dependiendo del problema

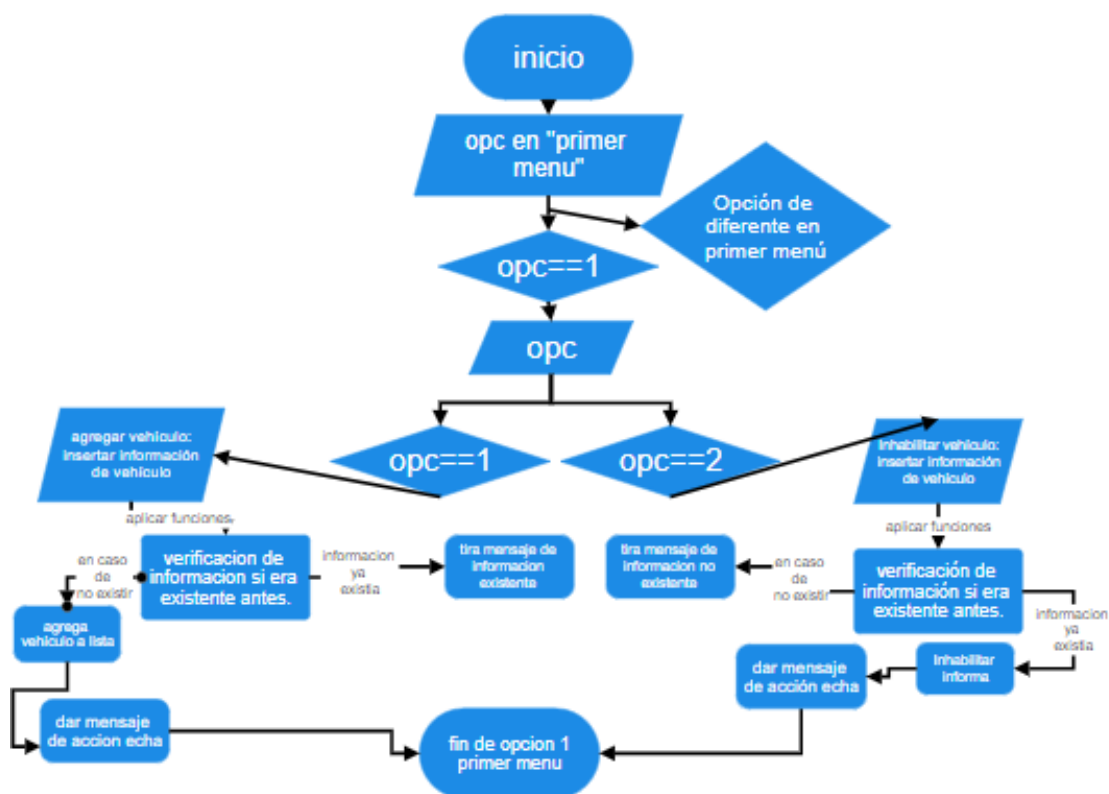
- Definir acciones dependiendo de la solicitud del usuario
 - Se dio solución a este problema con el uso de condicionales ifs los cuales evalúan la información otorgada en el input donde se pregunta al usuario la opción que desea y este escribirá una respuesta que coincida con las opciones programadas.
- Existencia de los elementos que se desean buscar, inhabilitar o reservar
 - Esto es referente a si los elementos y su conjunto respectivo que se necesitan están en las condiciones necesarias (existentes o no), se maneja este problema con (igualmente) el uso de "ifs" pero esta vez aplicándolas de forma que evalúen la existencia en una lista donde se almacenan todas las variables entre estas buscando la variable principal.

1.3 Algoritmo

- Pasos a seguir dependiendo de las condiciones y objetivo principal
 1. Pedir opción en menú principal
 2. comparo la opción de menú principal
 3. dar opciones de que tipo de gestión desea realizar
 4. comparo la opción de sub-menú
 5. (en caso de decisión de agregar vehículo) pedir información del vehículo
 6. Insertar información completa del vehículo
 7. Llamado de función
 8. Revisión de información si es existente previamente o no
 9. Agregar vehículo
 10. Devolver información actual del vehículo
 11. (en caso de decisión de inhabilitar vehículo) pedir información primordial del vehículo
 12. Insertar información primordial del vehículo
 13. Llamado de función
 14. Revisión de información si es existente previamente o no
 15. Conversión de dato a falso
 16. Devolver información actual del vehículo

1.4 Diagrama

(Con propósito de mejor comprensión para el lector)



2. Ingreso como cliente registrado

2.1 Diseño

```
elif opc == "2":
    print("Inhabilitar vehiculos")
    print("----Menu-----")
    print("[1] Vehiculo dañado ")
    print("[2] Vehiculo reservado")

elif opc == "2":
    print("Gestion de clientes")
    print("----Menu-----")
    print("[1] Ingresar como invitado")
    print("[2] Ingresar como cliente registrado")
    cli_opc = input("Seleccione una opción para la gestión de clientes: ")

    if cli_opc == "1":
        print("Cliente invitado")

    elif cli_opc == "2":
        print("Iniciar sesion")
        Nombre = input("Ingrese su nombre:")
        NumCedula = input("Ingrese su numero de cedula:")

        if NumCedula in Registro_Cedulas:
            print("Bienvenido de nuevo",Nombre)

        else:
            print("Cedula no encontrada, porfavor registrese")
            Registro_Cedulas.append(NumCedula)
            Nombre = input("Ingrese su nombre:")
            Telefono = input("Ingrese su numero de telefono:")
            print("Se registro correctamente ")
```

2.1.1 Códigos usados

2.1.1.1 Condición de Elif (se repite a lo largo del código)

elif opc == "2":	If/elif (condicion):
------------------	----------------------

2.1.1.1.1 Forma de aplicación y base

El propósito general del código “elif (condicional):” es dar una opción diferente de ejecución dependiendo de la condición que cumpla en el caso de que el “if (condicional):” no se haya cumplido este código como ya se sabe previamente.

En este código se utilizó para otorgar acciones a ejecutar, en este caso específico de que el usuario que este ejecutando el código haya declarado un “2” como variable “opc”.

se aplica a lo largo con diferentes variables y diferentes tipos de datos

Esta forma de código se repite a lo largo también realizando “if-anidados” que luego vamos a explicar.

2.1.1.1.2 Propósito

El propósito de usar este código, es poder ejecutar una serie de pasos específicos para poder darle al usuario mas opciones elegir en base a la información que declare en la

variable, un ejemplo del como se aplica este código es en la división de caminos en la sección declarada "gestión de clientes"(además de esta misma ser una de las opciones vigentes desde el primer menú dependiendo de la acción que el usuario quería ver o realizar) donde se proporcionan 2 opciones , lo cual dos alternativas que puede seguir el código, en una opción llamada "cliente invitado", en la cual el usuario después de declararse como cliente (escribiendo un "1" en las opciones) no realizara acción alguna (hasta el momento en este primer avance).

Si el usuario no se declara como cliente en el menú de "Gestión de clientes" y escribe un "2" se estará declarando como cliente registrado donde después se le solicitara ingresar información de "iniciar sesión".

Como se menciono antes se aplican varios "if" (if-anidados), esto se ve en la "confirmación de inicio cesión" , en la cual después de ingresar información se "inicia sesión", se ira a realizar la condicional que confirma que la variable "cedula" se encuentra afirmativamente en la lista de variables que si están ingresadas(esta variable se encuentra entre los datos principales de cada usuario que ya estuvo registrado y es la que define si la información ya estaba agregada o se necesita realizar un registro). De aquí se parten 2 caminos (realizado por otro if).

En en forma resumida referente al código "if", si el usuario (la variable correspondiente a la cedula) si se encuentra entre las variables de la lista de ya ingresados, se da un mensaje que le dice su estado en el programa (ya estaba registrado).

En caso de que no sea asi y no se encuentre la variable se aplica la condicional emergente ("else"), donde se piden datos que luego serán agregados a la lista de usuarios ingresados.

2.1.1.2 Lista (se repite en varias ocasiones del codigo)

Registro_Cedulas = []	Variable =[0, 1, 2, 3]
------------------------	-------------------------

2.1.1.2.1 Forma de aplicación y base

Este código se utiliza para agrupar un grupo de variables tanto de diferentes tipos como del mismo en una variable extra que guarde todos esos datos formando en el conjunto una "lista".

Es bien sabido que son las mas usadas y manipuladas al momento de programar, en ella se puede encontrar un dato especifico, realizar un recorrido de los datos, de paso de estos sumarlos, mostrarlos, editarlos al mismo tiempos etc.

Estas variables se caracterizan también porque sus elementos se etiquetan por su posición iniciando en cero (0), continuando con los números consecutivos de uno en uno, sin limite especifico.

En esta sección del código, su uso fue de almacenamiento de los datos que fueron pedidos en la cuarta condicional con respecto a la sección de "gestión cliente", luego dependiendo si la información se encontraba en la variable lista o no se elegiría una acción predeterminada(ya explicada previamente).

2.1.1.2.2 Propósito

Su propósito en el código (esta sección) es tanto: Limitar las acciones, definir si se tienen que registrar personas, verificar información que dependiendo de esta se tomaran diferentes caminos, recorrer información para tanto buscar como mostrar información del cliente después de su registro o al momento de iniciar sesión y lo que se considera mas importante y que definirá los pasos esenciales es el insertar información en la lista ubicada en "registro cliente".

2.1.1.3 Ingresar datos en variables (usado en todo el código)

Nombre = input("Ingrese su nombre:")	Variable = input(Informacion)
---	----------------------------------

2.1.1.3.1 Forma de aplicación y base

La descripción de este código es sencilla, se basa en un código cuyo único propósito es almacenar información, esta se define automáticamente como tipo "texto", si se desea cambiar a tipo número se necesitaría del código que funciona como convertidor "variable=int(variable)", automáticamente transforma las variables en números.

La información se puede manipular, reasignar y estas variables de almacenamiento se pueden asignar el mismo valor a otras variables, de la misma forma almacenamiento para compartir o trasladar momentáneamente como uno de sus usos.

2.1.1.3.2 Propósito

El propósito de este código, se extiende en: almacenar la información de los clientes, ser un medio a la hora de definir las tomas de acciones determinadas por las condiciones en las que también se usa la variable de almacenamiento, en esta área de código, ser un medio de hallazgo de información entre listas.

2.2 Planeamiento

2.2.1 identificación del problema

en los problemas que se ven a lo largo del código están.

- Definir acción dependiendo de la elección del usuario
- Condiciones a aplicar
- Enlistar información
- Verificar información dada

2.2.2 acciones dependiendo del problema

- Definir acción dependiendo de la elección del usuario
 - Aplicación de condicionales "if/if-elif/if-anidado"
 - También llamados como "menus"
- Condiciones a aplicar
 - Definido con "if" con información específica que evalúa la información de la variable
- Enlistar información
 - Aplicación de listas con conjunto de variables previamente ingresadas
- Verificar información dada
 - Arreglado con "if" con condiciones múltiples que evalúan la existencia de la información, en el caso de que ya existe previamente o no

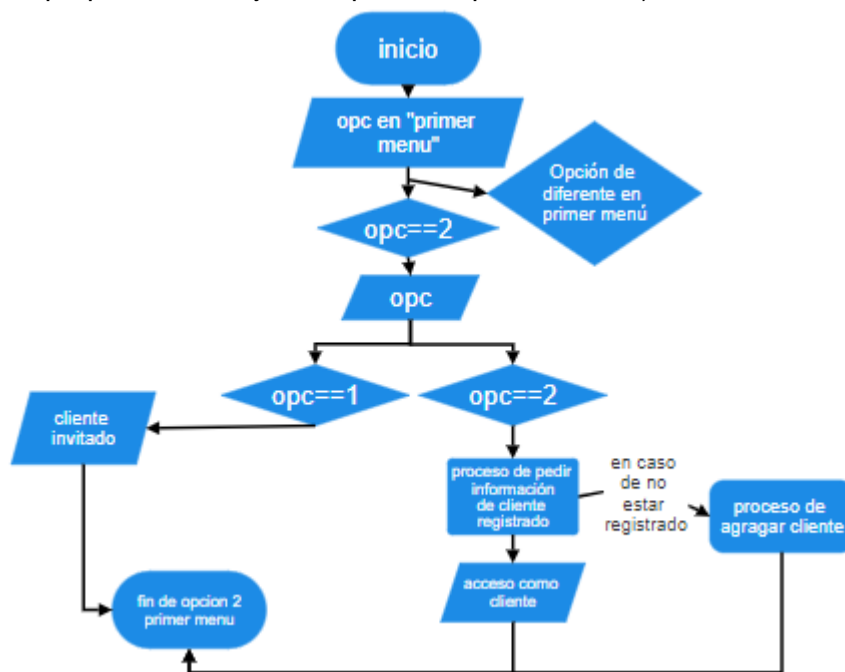
2.3 Algoritmo

- Pasos a seguir dependiendo de las condiciones y objetivo principal
 1. Pedir opción en menú principal
 2. Comparo la opción de menú principal
 3. pedir tipo de usuario

4. analizar tipo de usuario
5. definir acciones por tipo de usuario
6. usuario es invitado
7. usuario invitado dar mensaje
8. usuario es registrado actual
9. solicitar información del usuario registro
10. buscar información del usuario en registros
11. usuario es registrado anteriormente (se encontraba en registro)
12. usuario se le da mensaje de bienvenida
13. usuario no es registrado(no se encuentra en registro)
14. solicitar información mas amplia
15. agregar información al registro
16. dar bienvenida al usuario nuevo

2.4 Diagrama

(Con propósito de mejor comprensión para el lector)



3 Visualizar vehículos

3.1 Diseño

```

elif opc == "3":
    print("Visualizar vehículos")
    for i in inventario_vehiculos:
        print(i)
    print("Estos son los vehiculos")
  
```

3.1.1 Código usados

3.1.1.1 Recorrido for

for i in inventario_vehiculo s: print(i)	for variableAlmacenamien to in rangoOconjunto de elementos accion
---	---

3.1.1.1.1 Forma de aplicación y base

Este código se aplica con un ciclo for, el cual se le da la variable lista y una variable emergente que proporciona donde van a pasar las variables para luego ser impresas en pantalla.

Esta variable emergente mencionada cambiara su contenido consecutivamente, para cambiar su impresión, ocupando cada elementos de la lista hasta el final.

3.1.1.1.2 Propósito

Su propósito es limitarse a mostrar el contenido de la lista correspondiente en orden y de forma continua hasta que se halla mostrado cada conjunto de información.

3.1.1.2 estructura if /if-elif/for

print(i)	print(contenido)
----------	------------------

3.1.1.2.1 Forma de aplicación y base

Este código puede usarse de forma que pueda mostrar valores numéricos, datos, estados de datos, sus tipos y lo más usado escribir mensajes en pantalla

En este código se aplica de forma que dentro del ciclo for el contenido a imprimir y que es cambiante se muestre al usuario hasta que termine el ciclo.

3.1.1.2.2 Propósito

Su propósito fue de mostrar los elementos en la lista dentro del recorrido de datos hasta que ya no haya datos por mostrar dicho en sencillas palabras.

Nota: en el código se solicita imprimir la “variable emergente”

Ningún código realizado en esta opción requiere de la autorización del usuario, se realizan automáticamente.

Y si no se encuentran datos dentro de la lista, el código simplemente no mostrara nada

3.2 Planeamiento

3.2.1 identificación del problema

- como recorrer los elementos
- como mostrar los elementos

3.2.2 acciones dependiendo del problema

- como recorrer los elementos
 - este problema se soluciono con un for combinado con la lista el cual se encargaría de pasar cada conjunto de elementos a la “variable emergente”
- como mostrar los elementos

estos elementos serán mostrados dentro del bucle for proporcionados por la variable lista de “inventario_vehiculos”.

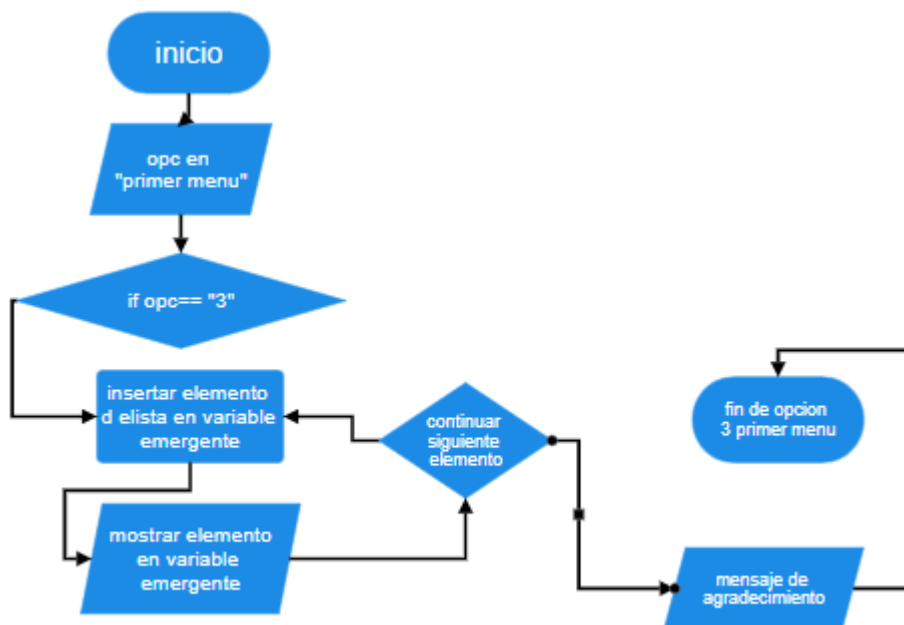
Explicando esto: una vez que el conjunto es pasado a la variable emergente, esta variable cambiaría cada vez que se recorra el ciclo hasta el final de la lista, por cada recorrido se usa un print que mostrara el contenido la variable emergente

3.3 Algoritmo

- Pasos a seguir dependiendo de las condiciones y objetivo principal
 1. Pedir opción en menú principal
 2. comparo la opción de menú principal
 3. entro a la indicada de opción 3
 4. inserto lista en ciclo con variable emergente
 5. insertar elemento en turno en la variable emergente
 6. muestro variable emergente
 7. repito acción 5 y 6 hasta ya no haber mas elementos
 8. mensaje y cierre del codigo

3.4 Diagrama

(Con propósito de mejor comprensión para el lector)



Cronograma

Avance	Requerimiento	Entrega
Avance I	Gestión inventario vehículos -Jordan Ivan Soto Morales	8 de marzo del 2024
	Ingreso como cliente registrado - Camila Argeñal Coronado	
	Mostrar vehículo -Eduardo miranda mora -Josias Aguilar Arroyo -Samantha Perez Rojas	
	Creación de repositorio ubicado en git hub -Jordan Ivan Soto Morales -Samantha perez rojas	
Avance II		27 de marzo del 2024
Avance III		17 de abril del 2024

