

SISTEMA DE CONTROL DE VEHÍCULOS

INSTRUCCIONES

Lee con atención cada uno de los requerimientos que se presentan a continuación, y desarrolla la prueba de acuerdo con lo solicitado.

DESCRIPCIÓN

El Equipo de desarrollo de software, se reúne con la finalidad de discutir lo referente al diseño de un sistema para el control de vehículos en un sistema de peaje. Según lo conversado con el cliente, se tienen los primeros datos:

1. Un vehículo contiene los siguientes atributos: marca, modelo y número de ruedas.
2. Un automóvil contiene los siguientes atributos: marca, modelo, número de ruedas, velocidad y cilindrada.

En este primer Sprint se acordó con el equipo de desarrollo las siguientes funcionalidades:

Para esta parte, dividen el sprint en:

Parte 1:

- Diseñe el diagrama de clases según los datos capturados con el cliente.
- Partiendo del diseño de diagrama de clases previamente construido, diseñe en las Clases en Python.
- Genere tres instancias, y al ejecutar el programa se debe mostrar lo siguiente:

Terminal

```
$ python main.py
Cuantos Vehiculos desea insertar: 3

Datos del automóvil 1
Inserte la marca del automóvil: Toyota
Inserte el modelo: Yaris
Inserte el número de ruedas 4
Inserte la velocidad en km/h: 120
Inserte el cilindraje en cc: 800

Datos del automóvil 2
Inserte la marca del automóvil: Fiat
Inserte el modelo: Palio
Inserte el número de ruedas 4
```

```
Inserte la velocidad en km/h: 95
Inserte el cilindraje en cc: 1200

Datos del automóvil 3
Inserte la marca del automóvil: Ford
Inserte el modelo: Fiesta
Inserte el número de ruedas 4
Inserte la velocidad en km/h: 125
Inserte el cilindraje en cc: 1500

Imprimiendo por pantalla los Vehículos:

Datos del automóvil 1 : Marca Toyota, Modelo Yaris, 4 ruedas 120 Km/h,
800 cc
Datos del automóvil 2 : Marca Fiat, Modelo Palio, 4 ruedas 95 Km/h,
1200 cc
Datos del automóvil 3 : Marca Ford, Modelo Fiesta, 4 ruedas 125 Km/h,
1500 cc
```

Parte 2:

Partiendo de la descripción anterior por parte del cliente, nos plantea que se manejan dos tipos de automóviles tipo: particular y carga, que contienen todas las características de un automóvil.

- Los automóviles tipo particular contienen adicionalmente los números de puesto.
- Los automóviles tipo carga contienen adicionalmente el peso de la carga en kg.

Adicionalmente, se tienen el tipo de vehículos que son bicicleta que contiene las características de los vehículos, y se le adiciona el tipo de bicicleta que puede ser: Urbana o de Carrera.

Con los tipos de bicicletas tenemos las motocicletas que contienen todas las características de una bicicleta, además de las siguientes: nro_radios, cuadro y motor.

```
nro_radios: 21 radios
# cuadro: doble cuna, multitubolar, doble viga
# motor: 2T o 4T
```

Actividades en la parte 2

- Diseñe el Diagrama de Clases para las exigencias antes mencionadas por parte del cliente.
- Codifique en Python las clases previamente diseñadas.
- Agregue las siguientes instancias en el archivo principal.

```
1 particular = Particular("Ford", "Fiesta", 4, "180", "500", 5)
2 carga = Carga("Daft Trucks", "G 38", 10, 120, "1000", "20000")
3 bicicleta = Bicicleta("Shimano", "MT Ranger", 2, "Carrera")
4 motocicleta = Motocicleta("BMW", "F800s", 2, "Deportiva", "2T", "Doble
5 Viga", 21)
```

Salida:

```
$ python main.py
Marca Ford, Modelo Fiesta, 4 ruedas 180 Km/h, 500 cc Puestos: 5
Marca Daft Trucks, Modelo G 38, 10 ruedas 120 Km/h, 1000 cc Carga:
20000 Kg
Marca Shimano, Modelo MT Ranger, 2 ruedas Tipo: Carrera
Marca MW, Modelo KS, 2 ruedas Tipo: Deportiva Motor: 2T, Cuadro: Doble
Viga, Nro Radios: 21
```

Verificar la relación que existe de la instancia motocicleta con: Vehículo, Automóvil, Particular, Carga, Bicicleta y Motocicleta. Obteniendo la siguiente salida:

```
Motocicleta es instancia con relación a Vehículo: True
Motocicleta es instancia con relación a Automovil: False
Motocicleta es instancia con relación a Vehículo particular: False
Motocicleta es instancia con relación a Vehículo de Carga: False
Motocicleta es instancia con relación a Bicicleta: True
Motocicleta es instancia con relación a Motocicleta: True
```

Parte 3:

En esta tercera parte el cliente quiere que se guarden los datos en un archivo con el nombre de **vehiculos.csv**, es decir, un archivo separado por comas.

Para esto se debe tener en cuenta el siguiente ejemplo de guía con relación a la manipulación de objetos en los archivos **csv**:

```
1 from Vehiculo import Automovil
2 import csv
3
4 def guardar(nombre_archivo, Automovil):
5     archivo = open(nombre_archivo, "w")
6     datos = [(Automovil.__class__, Automovil.__dict__)]
7     archivo_csv = csv.writer(archivo)
8     archivo_csv.writerow(datos)
9     archivo.close()
10
11 def recuperar(nombre_archivo):
12     vehiculos = []
13     archivo = open(nombre_archivo, "r")
14     archivo_csv = csv.reader(archivo)
15     for vehiculo in archivo_csv:
16         vehiculos.append(vehiculo)
```

```

17     archivo.close()
18     return vehiculos
19
20 automovil = Automovil("Ford", "Fiesta", "4", "180", "500")
21
22 guardar("ejemplo.csv", automovil)
23
24 automoviles = recuperar("ejemplo.csv")
25
26 for automovil in automoviles:
27     print(automovil)

```

Salida del ejemplo:

```

$ python ejemplo_csv.py
[<class 'Vehiculo.Automovil'>,"{'marca': 'Ford', 'modelo':
'Fiesta', 'nro_ruedas': '4', 'velocidad': '180', 'cilindrada':
'500'}"]

```

Actividades en este parte 3:

- Crear el método que permita guardar cada uno de los objetos previamente creados en el archivo **vehiculos.csv**, el nombre del método es: **guardar_datos_csv(self)**

Tome como referencia los siguientes objetos:

```

1 particular = Particular("Ford", "Fiesta", "4", "180", "500", "5")
2 carga = Carga("Daft Trucks", "G 38", "10", "120", "1000", "20000")
3 bicicleta = Bicicleta("Shimano", "MT Ranger", 2, "Carrera")
4 motocicleta = Motocicleta("BMW", "F800s", 2, "Deportiva", "2T", "Doble
5 Viga", 21)

```

- En el archivo **vehiculos.csv** se guardan de la siguiente manera:

```

1 <class 'Vehiculo.Particular'>,"{'marca': 'Ford', 'modelo':
2 'Fiesta', 'nro_ruedas': '4', 'velocidad': '180', 'cilindraje':
3 '500', 'nro_puestos': '5'}"
4 <class 'Vehiculo.Carga'>,"{'marca': 'Daft Trucks', 'modelo': 'G
5 38', 'nro_ruedas': '10', 'velocidad': '120', 'cilindraje': '1000',
6 'carga': '20000'}"
7 <class 'Vehiculo.Bicicleta'>,"{'marca': 'Shimano', 'modelo': 'MT
8 Ranger', 'nro_ruedas': 2, 'tipo': 'Carrera'}"
9 <class 'Vehiculo.Motocicleta'>,"{'marca': 'BMW', 'modelo': 'F800s',
10 'nro_ruedas': 2, 'tipo': 'Deportiva', 'motor': '2T', 'cuadro':
11 'Doble Viga', 'nro

```

La estructura para guardar son la clase, y luego los atributos del objeto de la clase.

- Cree un método que lea del archivo **vehiculos.csv** con el nombre **leer_datos_csv(self)** y que imprima por pantalla o terminal lo siguiente según la clasificación del vehículo:

```
$ python main.py
Lista de Vehiculos Particular
{'marca': 'Ford', 'modelo': 'Fiesta', 'nro_ruedas': '4', 'velocidad':
'180', 'cilindrada': '500', 'nro_puestos': '5'}

Lista de Vehiculos Carga
{'marca': 'Daft Trucks', 'modelo': 'G 38', 'nro_ruedas': '10',
'velocidad': '120', 'cilindrada': '1000', 'carga': '20000'}

Lista de Vehiculos Bicicleta
{'marca': 'Shimano', 'modelo': 'MT Ranger', 'nro_ruedas': 2, 'tipo':
'Carrera'}

Lista de Vehiculos Motocicleta
{'marca': 'BMW', 'modelo': 'F800s', 'nro_ruedas': 2, 'tipo':
'Deportiva', 'motor': '2T', 'cuadro': 'Doble Viga', 'nro_radios': 21}
```

- En los métodos creados utilice el manejo de excepciones en caso de errores en el archivo.