

Nombre: juan camilo parra sanchez

ID: 917079

Requerimiento Funcional 1

Nombre	R1 – Registrar Camión
Resumen	Permite a la aplicación registrar un camión nuevo
Entradas	Matricula del Camión, Capacidad del camión(Kg), Consumo de Gasolina (galones/Kilometro), Carga Actual (Kg)
Resultado	La aplicación registra al camión nuevo si tiene la capacidad suficiente (4)

Requerimiento Funcional 2

Nombre	R2 – Cargar Camión
Resumen	Permite a la aplicación cargar un camión, verificando que el peso no exceda su capacidad máxima
Entradas	Matricula del Camión, Peso de la carga a añadir (Kg)
Resultado	Si el peso de la carga es menor o igual a la capacidad máxima del camión, la carga se añade, si la carga es mayor a la carga de camión, la operación es rechazada

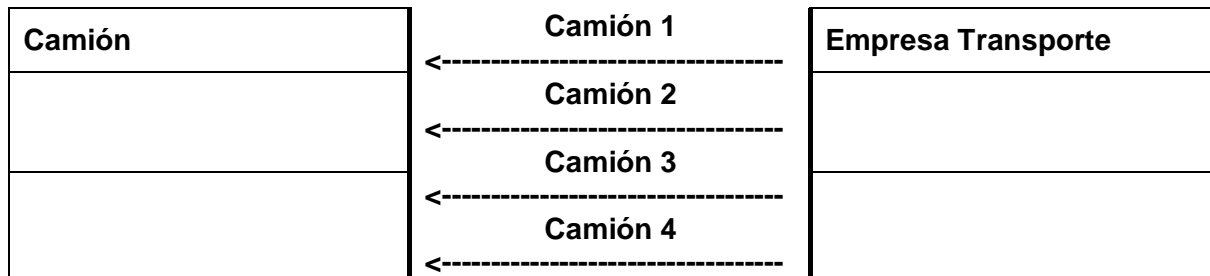
Requerimiento Funcional 3

Nombre	R3 – Descargar Camión
Resumen	Permite a la aplicación descargar una cantidad del camión
Entradas	Matricula del Camión, Peso de la carga a retirar
Resultado	Se descarga la cantidad especificada del camión

Requerimiento Funcional 4

Nombre	R4 – Determinar el Mejor camión disponible
Resumen	El programa debe determinar cuál es el camión con menor consumo de gasolina y suficiente capacidad de carga
Entradas	Peso de la carga a transportar (Kg)
Resultado	El programa muestra el camión con menor consumo de gasolina y la capacidad de carga requerida, si ningún camión cumple estos requisitos el programa informa que no hay ningún camión disponible

Clases



Implementación Métodos Python

```
class camion:

    # -----
    # Metodos
    # -----

    __method__ = "darCapacidad"
    __params__ = "Ninguno"
    __returns__ = "Capacidad maxima de carga de el camion en Kg"
    __descriptions__ = (
        "retorna la capacidad de carga maxima del camion especificado en
Kg"
    )

    def darCapacidad(self):
        return self.capacidad

    __method__ = "darConsumo"
    __params__ = "Ninguno"
    __returns__ = "El consumo de gasolina en galones por kilometro "
    __descriptions__ = (
        "retorna el consumo de gasolina del camion en galones por kilometro "
    )

    def darConsumo(self):
        return self.consumo

    __method__ = "darMatricula"
    __params__ = "Ninguno"
    __returns__ = "Matricula del camion"
    __descriptions__ = (
        "Retorna la matricula del camion seleccionado, el cual es su
identificador"
```

```

)

def darMatricula(self):
    return self.Matricula

__method__ = "darCargaActual"
__params__ = "Ninguno"
__returns__ = "Carga actual del camion"
__descriptions__ = "Retorna la carga actual que el camion esta
transportando en kg"

def darCargaActual(self):
    return self.cargaActual

__method__ = "Cargar"
__params__ = "carga"
__returns__ = "True Cuando se añade la carga, False si la carga excede la
capacidad del camion"
__descriptions__ = "Añade una carga especifica al camion, si esta no
excede la capacidad maxima del mismo"

def Cargar(self, carga):
    self.cargaActual += (carga + self.cargaActual <= self.capacidad) *
carga
    return self.cargaActual + carga <= self.capacidad

__method__ = "Descargar"
__params__ = "carga"
__returns__ = "True si la descarga fue exitosa, False si no hay suficiente
carga que descargar "
__descriptions__ = "descarga una cantidad especifica del camion"

def Descargar(self, carga):
    self.cargaActual -= (self.cargaActual >= carga) * carga
    return self.cargaActual >= carga

```