



# TRABAJO PRÁCTICO 2

PROGRAMACIÓN 2

PROFESOR: GUERRA, MARIO

ALUMNO: BUTASSI, FEDERICO

## P2 Trabajo Práctico N.º 2

# Modelado de Clases

## Presentación

En este trabajo práctico se realizará un ciclo abreviado del diseño e implementación de un sistema simple para una empresa dedicada al alquiler de automóviles, partiendo de una lista de requerimientos, y logrando el modelado e implementación de las clases en un entorno de programación y las entidades necesarias para una base de datos relacional.

## Objetivos

- Modelar el problema logrando un diagrama estático de clases
- Implementar dichas clases en un IDE con el lenguaje de programación Java
- Modelar las entidades necesarias para almacenar en una base de datos

Este trabajo práctico articula contenidos de los espacios curriculares “Modelado de Sistemas”, “Programación II” y “Bases de Datos” del 2º año de la Tecnicatura. Por el lado del primero se trabajan los contenidos “Análisis orientado a objetos y UML” y “Diagramas de clase”. Por el lado del segundo se trabajan los contenidos “Clases y encapsulamiento”. Del tercero se trabajan los contenidos “Modelización de datos, modelo relacional”.

## Enunciado

Se trabajará sobre la información de las reservas de una empresa dedicada al alquiler de automóviles, teniendo en cuenta las siguientes premisas:

- Un determinado cliente puede tener en un momento dado hechas varias reservas.
- De cada cliente se desea almacenar su DNI, nombre, dirección y teléfono. Además dos clientes se diferenciarán por un código único.
- Cada cliente puede ser avalado por otro cliente de la empresa.
- Una reserva la realiza un único cliente pero puede involucrar varios autos.
- Es importante registrar la fecha de inicio y final de la reserva, el precio del alquiler de cada uno de los autos, los litros de nafta en el depósito de cada auto en el momento de realizar la reserva, el precio total de la reserva y un indicador de si el auto o los autos han sido ya entregados.
- Todo automóvil tiene siempre asignado un determinado garaje que no puede cambiar. De cada auto se requiere la matricula, el modelo el color y la marca.
- Cada reserva se realiza en una determinada agencia.

## Desarrollo

Se propone realizar el modelado de las clases usando el programa StarUML con "Class Diagram". Opcionalmente se puede presentar alguno de los casos de uso. Se deberán agregar al menos 2 métodos para alguna de las clases. Para la programación se debe usar el programa NetBeans, donde todos los atributos deberán ser tratados como encapsulados, generando los correspondientes getters/setters. Se deben incluir constructores en cada clase, donde se inicialicen los atributos y se emita un mensaje. Se debe incluir un programa principal donde se instancien objetos de todas las clases, y se llamen a los métodos (el cuerpo de los métodos solo debe imprimir un mensaje acorde). Para el modelo de entidad/relación se sugiere usar el programa StarUML con "ER Diagram". De los diagramas exportar imágenes para incluirlas en el documento final, y del programa Java presentar el fuente de cada clase en el documento final.

## Criterios de evaluación

El trabajo deberá presentarse en formato PDF con tamaño de hoja A4. Para evaluar se realizará una rúbrica donde se tendrá en cuenta: La presencia de carátula y prolijidad de la presentación, la claridad de los textos donde se explique el trabajo efectuado, la existencia de todos los items solicitados y la completitud general del trabajo.

## Recursos

- Vídeo "Diagrama de clases y diagramas de casos de uso en StarUML"  
<https://www.youtube.com/watch?v=VY4zQFudxsc>
- Vídeo "Encapsulamiento y métodos accesorios (Getters y Setters)"  
<https://www.youtube.com/watch?v=8LyjCiuyNkw>
- Vídeo "Crear constructor, getter y setter automáticamente en Netbeans"  
<https://www.youtube.com/watch?v=WSS2orEGfJo>
- Vídeo diagramas Entidad-Relación "StarUML- ER"  
<https://www.youtube.com/watch?v=LSEJNz4ZKqk>

# ÍNDICE

INTRODUCCIÓN	2
DIAGRAMA DE CLASES	3
CODIGO FUENTE	4-13

**Alumno:** Butassi Federico

**Fecha:** 06/07/2023

**Profesor:** Guerra Mario

## **1) INTRODUCCIÓN**

Para comenzar, comencé utilizando el programa starUML para crear un diagrama de clases que luego me ayudaría a desarrollar según la consigna asignada.

En primera instancia, cree las clases correspondientes en NetBeans, con sus variables y constructor respectivamente para cada una de ellas.

Luego, crea los métodos getters y setters para cada atributo. Luego buscando seguir con la idea de proyecto que tenía en mente opte por utilizar algunas clases de tipo enum, como por ejemplo en las clases, “Auto”, “Reserva” y “Cliente”.

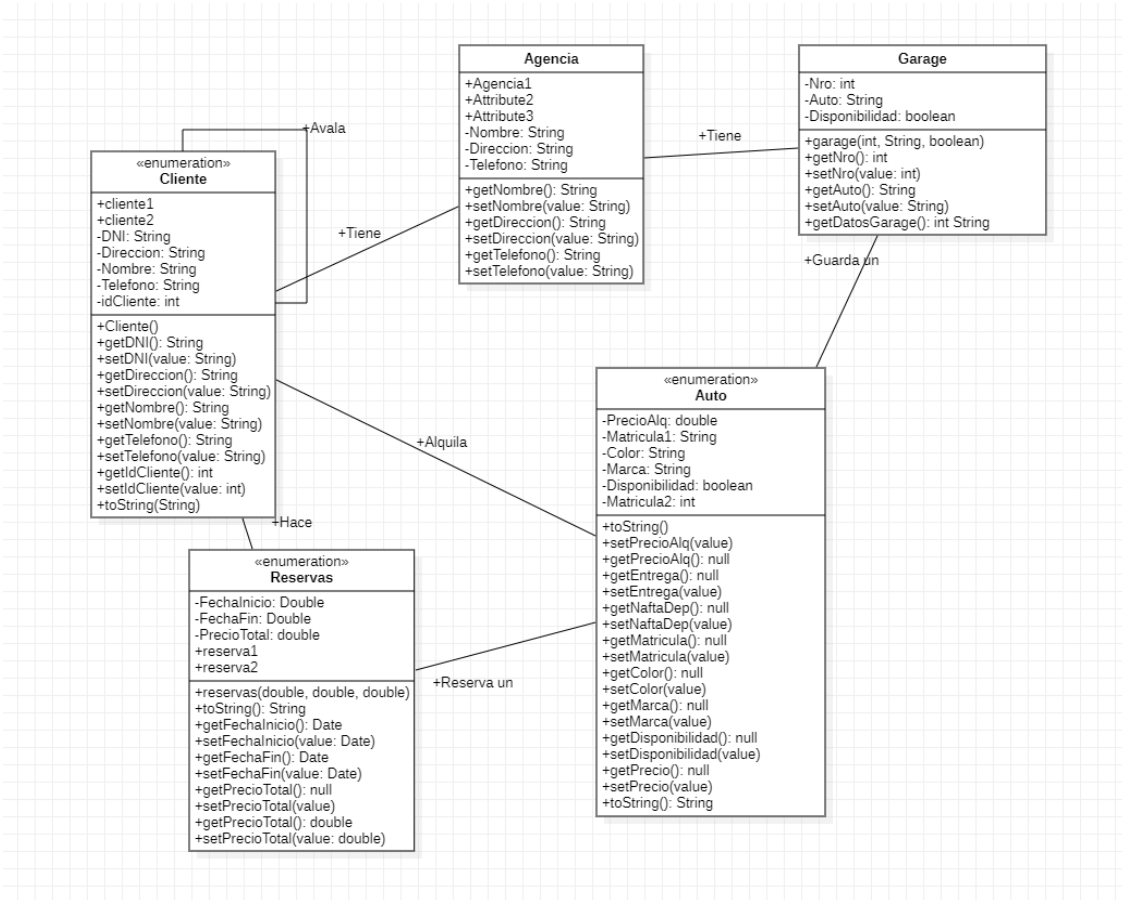
Por último, vale aclarar que mi idea de proyecto se basaba en que un cliente entra al sistema buscando alquilar un automóvil. Allí se le mostrará un mensaje detallando las diferentes agencias que tiene disponible en la zona. Las cuales tienen dos garajes cada una, y al cliente se le muestra que auto está disponible, con su valor, su marca, su color y su matrícula.

También, se le enseña algunos garajes donde no tenía ningún auto disponible y se le muestra quien reservó el auto, los detalles del auto, y cuando iba a ser devuelto y por que cliente había sido alquilado.

Después, se le ofrecen los autos disponibles, el decide con total libertad cuál decide alquilar, se le pregunta si quiere alquilar algún otro, cuando dice que no, se le hace llenar un formulario con sus datos personales.

Y por último un aviso de “TRANSACCIÓN EXITOSA” en donde se lo felicita por su reserva.

## 2) DIAGRAMA DE CLASES:



### 3) CÓDIGO FUENTE:

```
1 package com.mycompany.principal;
2
3
4 import java.util.Scanner;
5
6 public class Principal {
7
8     public static void main(String[] args) throws InterruptedException {
9         Agencia[] agencias = Agencia.values();
10        System.out.println("AGENCIAS DISPONIBLES EN LA ZONA");
11        System.out.println("");
12        for(int i = 0; i < agencias.length; i++){
13            System.out.println((i + 1) + ") " + agencias[i]);
14        }
15        System.out.println("");
16        System.out.println("Cada agencia en cuestion, tiene un total de 2(dos) garages");
17        System.out.println("-----");
18        System.out.println(" " + Agencia.agencia1+ "\n" + "Tiene solo uno de sus dos lugares disponibles para alquilar.");
19
20        Auto[] autos = Auto.values();
21        System.out.println("");
22        for (int i = 0; i < autos.length-4; i++) {
23            System.out.println((i+1) + " " + autos[i]);
24        }
25        System.out.println("-----");
26        System.out.println(" " + Agencia.agencia1+ "\n" + "Tiene dos de sus lugares disponibles para alquilar.");
27        System.out.println("");
28        for (int i = 2; i < autos.length-2; i++) {
29            System.out.println((i+1) + " " + autos[i]);
30        }
31        System.out.println("");
32        System.out.println("-----");
33        System.out.println("");
34        System.out.println(" " + Agencia.agencia1+ "\n" + "Tiene solo uno de sus dos lugares disponibles para alquilar.");
35        System.out.println("");
36        for (int i = 4; i < autos.length; i++) {
37            System.out.println((i+1) + " " + autos[i]);
38        }
39        System.out.println("////////////////////////");
40        System.out.println("Estos autos no estan disponibles: ");
41
42        for (Auto auto : Auto.values()) {
43            if (!auto.disponibilidad) {
44                System.out.println(" " + auto);
45            }
46        }
47        System.out.println("Alquilados por los siguientes clientes");
48        System.out.println("");
49        System.out.println("Auto: " + auto2.getMarca() + ") Fue entregado el dia " + Reservas.getReserva1().getFechaInicio());
50        System.out.println("Sera devuelto el dia " + Reservas.getReserva1().getFechaFin());
51        System.out.println("Por un valor total de: " + Reservas.reserva1.getPreciototal());
```

```

52     Cliente[] clientes = Cliente.values();
53     for (int i = 0; i < clientes.length - 1; i++) {
54         System.out.println((i+1) + "-" + clientes[i]);
55     }
56     System.out.println("\n");
57     System.out.println(Auto.auto5.getMarca() + ") Fue entregado el dia " + Reservas.getReserva2().getFechaInicio());
58     System.out.println(" Sera devuelto el dia " + Reservas.getReserva2().getFechaFin());
59     System.out.println(" Por un valor total de: " + Reservas.reserva2.getPrecioTotal());
60     for (int i = 1; i < clientes.length; i++) {
61         System.out.println((i+1) + "-" + clientes[i]);
62     }
63     System.out.println("\n-----");
64     System.out.println("\nDesea alquilar alguno de estos otros?");
65     System.out.println();
66     for (Auto auto : Auto.values()) {
67         if (auto.disponibilidad) {
68             System.out.println("\n" + auto);
69         }
70     }
71
72     Scanner entrada = new Scanner(System.in);
73     char respuesta;
74     int opcion;
75
76     do {
77         do {
78             System.out.println("\nQué auto desea?");
79             System.out.println("1. " + Auto.auto1.getMarca());
80             System.out.println("2. " + Auto.auto3.getMarca());
81             System.out.println("3. " + Auto.auto4.getMarca());
82             System.out.println("4. " + Auto.auto6.getMarca());
83             System.out.print("\nOpción: ");
84             opcion = entrada.nextInt();
85         } while (opcion < 1 || opcion > 4);
86
87         switch (opcion) {
88             case 1:
89                 System.out.println("Usted seleccionó: " + Auto.auto1);
90                 break;
91             case 2:
92                 System.out.println("Usted seleccionó: " + Auto.auto3);
93                 break;
94             case 3:
95                 System.out.println("Usted seleccionó: " + Auto.auto4);
96                 break;
97             case 4:
98                 System.out.println("Usted seleccionó: " + Auto.auto6);
99                 break;
100        }

```



```

100     }
101
102     System.out.print("\nDesea seleccionar otro auto (SI/NO)? ");
103     respuesta = entrada.next().charAt(index: 0);
104     System.out.println();
105 } while (respuesta == 's' || respuesta == 'S');
106 boolean ejecutarBucle = true;
107 boolean datosCompletos = false;
108
109 do{
110
111     System.out.println("Ingrese sus datos personales para confirmar la transaccion: ");
112     System.out.println("Nombre: ");
113     String nombre;
114     nombre = entrada.next();
115
116     System.out.println("Apellido: ");
117     String apellido;
118     apellido = entrada.next();
119
120     System.out.println("DNI: ");
121     String dni;
122     dni = entrada.next();
123
124     System.out.println("Direccion: ");
125     String direccion;
126     direccion = entrada.next();
127
128     System.out.println("Telefono: ");
129     String telefono;
130     telefono = entrada.next();
131     System.out.println("-----TRANSACCION CONFIRMADA-----");
132
133     switch (opcion) {
134
135     case 1:
136         System.out.println("Enhorabuena " + nombre + " " + apellido + "\n" +
137             "HA RESERVADO EL AUTO: " + "\n" + " " +
138             opcion + " " + Auto.auto1);
139
140         break;
141     case 2:
142         System.out.println("Enhorabuena " + nombre + " " + apellido + "\n" +
143             "HA RESERVADO EL AUTO: " + "\n" + " " +
144             opcion + " " + Auto.auto3);

```

```

145         break;
146     case 3:
147         System.out.println("Enhorabuena " + nombre + " " + apellido + "\n" +
148             "HA RESERVADO EL AUTO: " + "\n" + " " +
149             opcion + " " + Auto.auto4);
150
151         break;
152     case 4:
153         System.out.println("Enhorabuena " + nombre + " " + apellido + "\n" +
154             "HA RESERVADO EL AUTO: " + "\n" + " " +
155             opcion + " " + Auto.auto6);
156         break;
157     }
158
159 }
160
161 // verificar si los datos están completos
162 if (!nombre.isEmpty() && !apellido.isEmpty() && !dni.isEmpty() && !direccion.isEmpty() && !telefono.isEmpty()) {
163     datosCompletos = true;
164 }
165
166 // verificar si se deben interrumpir los bucles
167 if (datosCompletos || respuesta == 'n' || respuesta == 'N') {
168     ejecutarBucle = false;
169 }
170
171 } while (ejecutarBucle);
172
173 }
174

```

```

1 package com.mycompany.principal;
2
3
4 public class Garage {
5     private int nro;
6     private String auto;
7     boolean disponibilidad;
8
9     public Garage(int nro, String auto, boolean disponibilidad) {
10         this.nro = nro;
11         this.auto = auto;
12         this.disponibilidad = disponibilidad;
13     }
14
15     public void garages(int nro, String auto){
16
17     }
18     public int getNro() {
19         return nro;
20     }
21
22     public String getAuto() {
23         return auto;
24     }
25
26     public void setNro(int nro) {
27         this.nro = nro;
28     }
29
30     public void setAuto(String auto) {
31         this.auto = auto;
32     }
33     public String getDatosGareges(int nro, String auto){
34         return "Garages: " + nro + auto;
35     }
36
37 }
38

```

```

1 package com.mycompany.principal;
2
3
4 public enum Reservas {
5     reserva1(fechainicio:11.12,fechafin: 12.01, preciototal: 12500),
6     reserva2(fechainicio:10.11,fechafin: 10.12,preciototal: 11200);
7     double fechainicio;
8     double fechafin;
9     double preciototal;
10
11     private Reservas(Double fechainicio, Double fechafin, double preciototal) {
12         this.fechainicio = fechainicio;
13         this.fechafin = fechafin;
14         this.preciototal = preciototal;
15     }
16
17     public static Reservas getReserva1() {
18         return reserva1;
19     }
20
21     public static Reservas getReserva2() {
22         return reserva2;
23     }
24
25     public double getFechainicio() {
26         return fechainicio;
27     }
28
29     public double getFechafin() {
30         return fechafin;
31     }
32
33     public double getPreciototal() {
34         return preciototal;
35     }
36
37     @Override
38     public String toString() {
39         return "Reservas[" + "fechainicio=" + fechainicio + ", fechafin=" + fechafin + ", preciototal=" + preciototal + "]";
40     }
41
42 }

```

```

1
2 package com.mycompany.principal;
3
4 public class ReservaNueva {
5     private final int dni;
6     private final String direccion;
7     private final String nombreadellido;
8     private final String telefono;
9     private final int idCliente;
10
11
12     public ReservaNueva(int dni, String direccion, String nombreadellido, String telefono, int idCliente) {
13         this.dni = dni;
14         this.direccion = direccion;
15         this.nombreadellido = nombreadellido;
16         this.telefono = telefono;
17         this.idCliente = idCliente;
18     }
19
20     public int getDni() {
21         return dni;
22     }
23
24     public String getDireccion() {
25         return direccion;
26     }
27
28     public String getNombreadellido() {
29         return nombreadellido;
30     }
31
32     public String getTelefono() {
33         return telefono;
34     }
35
36     public int getIdCliente() {
37         int numero = (int)(Math.random()*(200-10+1)+10);
38         return idCliente;
39     }
40
41     @Override
42     public String toString() {
43         return "ReservaNueva[" + "dni=" + dni + ", direccion=" + direccion + ", nombreadellido=" + nombreadellido + ", telefono=" + telefono + ", idCliente=" + idCliente + "]";
44     }
45 }
46

```

```

1 package com.mycompany.principal;
2
3
4 public enum Auto {
5     auto1(precioAuto: 3550, naftaDep: 12.4, matricula1: "NXT", matricula2: 345, color: "Rojo", marca: "Mazda", disponibilidad: true),
6     auto2(precioAuto: 2230, naftaDep: 11.5, matricula1: "HWP", matricula2: 352, color: "Verde", marca: "Nissan", disponibilidad: false),
7     auto3(precioAuto: 2230, naftaDep: 14.2, matricula1: "PIO", matricula2: 569, color: "Negro", marca: "Nissan", disponibilidad: true),
8     auto4(precioAuto: 4539, naftaDep: 16.5, matricula1: "FTO", matricula2: 469, color: "Negro", marca: "Ferrari", disponibilidad: true),
9     auto5(precioAuto: 3459, naftaDep: 12.5, matricula1: "TRO", matricula2: 589, color: "Blanco", marca: "Peugeot", disponibilidad: false),
10    auto6(precioAuto: 1890, naftaDep: 10.5, matricula1: "POO", matricula2: 789, color: "Rojo", marca: "Fiat", disponibilidad: true);
11
12    private final int precioAuto;
13    private final double naftaDep;
14    private final String matricula1;
15    private final int matricula2;
16    private final String color;
17    private final String marca;
18    final boolean disponibilidad;
19
20    private Auto (int precioAuto, double naftaDep, String matricula1, int matricula2, String color, String marca, boolean disponibilidad) {
21        this.precioAuto = precioAuto;
22        this.naftaDep = naftaDep;
23        this.matricula1 = matricula1;
24        this.matricula2 = matricula2;
25        this.color = color;
26        this.marca = marca;
27        this.disponibilidad = disponibilidad;
28    }
29
30    public String getColor() {
31        return color;
32    }
33
34    public String getMarca() {
35        return marca;
36    }
37
38    public static Auto getAuto1() {
39        return auto1;
40    }
41
42    public static Auto getAuto2() {
43        return auto2;
44    }
45
46    public static Auto getAuto3() {
47        return auto3;
48    }
49
50    public static Auto getAuto4() {
51        return auto4;
52    }
53
54    public static Auto getAuto5() {
55        return auto5;
56    }
57
58    public static Auto getAuto6() {
59        return auto6;
60    }
61
62    public String getDisponibilidadString() {
63        return disponibilidad ? "Disponible" : "No disponible";
64    }
65
66    @Override
67    public String toString(){
68        return "Auto: " + "$" + precioAuto + " - " + "Nafta disponible: " + naftaDep + " - " + "Matricula: " + " " + matricula1 + matricula2 + "\n" + "Color: " + color + " - " + "Marca: " + marca + "\n" + getDisponibilidadString();
69    }
70
71 }
72

```

```

1
2 package com.mycompany.principal;
3
4 public enum Cliente {
5     cliente1(idCliente: 00234, nombreadellido: "Butassi Federico", dni: 41280335, direccion: "Espania 348", telefono: "3571327923"),
6     cliente2(idCliente: 00563, nombreadellido: "Jeff Bezos", dni: 31023892, direccion: "Dean Funes 1244", telefono: "3571890293");
7
8     private final int dni;
9     private final String direccion;
10    private final String nombreadellido;
11    private final String telefono;
12    private final int idCliente;
13
14    private Cliente(int idCliente, String nombreadellido, int dni, String direccion, String telefono) {
15        this.idCliente = idCliente;
16        this.dni = dni;
17        this.direccion = direccion;
18        this.nombreadellido = nombreadellido;
19        this.telefono = telefono;
20    }
21
22    @Override
23    public String toString(){
24
25        return "Cliente: id" + idCliente + "\n" + nombreadellido + "\n" + "DNI: " + dni + " - " + "Direccion: " + direccion + "\n" + "Telefono: " + telefono ;
26    }
27
28 }
29
30

```

```

1
2 package com.mycompany.principal;
3
4 public enum Agencia {
5     agencia1(nombre: "AG automotores", direccion: "Espania 1234", telefono: "3571238923"),
6     agencia2(nombre: "Gioda automotores", direccion: "Savio 125", telefono: "3571328920"),
7     agencia3(nombre: "Petiti", direccion: "Dean Funes 345", telefono: "3571283922");
8
9     private final String nombre;
10    private final String direccion;
11    private final String telefono;
12
13    private Agencia(String nombre, String direccion, String telefono) {
14        this.nombre = nombre;
15        this.direccion = direccion;
16        this.telefono = telefono;
17    }
18
19
20
21    @Override
22    public String toString(){
23
24        return "Agencia: " + nombre + " - " + "Direccion: " + direccion + " - " + "Telefono de contacto: " + telefono;
25    }
26
27 }
28
29
30

```

```

Scanning for projects...

-----< com.mycompany:Principal >-----
[+] Building Principal 1.0-SNAPSHOT
    from pom.xml
-----[ jar ]-----

[+] --- resources:3.3.0:resources (default-resources) @ Principal ---
- skip non existing resourceDirectory C:\Users\mercadito\OneDrive\Documentos\NetBeansProjects\Principal\src\main\resources

[+] --- compiler:3.10.1:compile (default-compile) @ Principal ---
- Nothing to compile - all classes are up to date

[+] --- exec:3.1.0:exec (default-cli) @ Principal ---
AGENCIAS DISPONIBLES EN LA ZONA

1)Agencia: AG automotores - Direccion: Espania 1234 - Telefono de contacto: 3571238923
2)Agencia: Gioda automotores - Direccion: Savio 125 - Telefono de contacto: 3571328920
3)Agencia: Petiti - Direccion: Dean Funes 345 - Telefono de contacto: 3571283922

Cada agencia en cuestion, tiene un total de 2(dos) garages
-----
-Agencia: AG automotores - Direccion: Espania 1234 - Telefono de contacto: 3571238923
Tiene solo uno de sus dos lugares disponibles para alquilar.

1-Auto: $3550 - Nafta disponible: 12.4 - Matricula: NXT345
Color: Rojo - Marca: Mazda
Disponible
2-Auto: $2230 - Nafta disponible: 11.5 - Matricula: HWP352
Color: Verde - Marca: Nissan
No disponible
-----
-Agencia: AG automotores - Direccion: Espania 1234 - Telefono de contacto: 3571238923
Tiene dos de sus lugares disponibles para alquilar.

3-Auto: $2230 - Nafta disponible: 14.2 - Matricula: PIO569
Color: Negro - Marca: Nissan
Disponible
4-Auto: $4539 - Nafta disponible: 16.5 - Matricula: FTO469
Color: Negro - Marca: Ferrari
Disponible

```

```
-----  
-Agencia: AG automotores - Direccion: Espania 1234 - Telefono de contacto: 3571238923  
Tiene solo uno de sus dos lugares disponibles para alquilar.  
  
5-Auto: $3459 - Nafta disponible: 12.5 - Matricula: TRO589  
Color: Blanco - Marca: Peugeot  
No disponible  
6-Auto: $1890 - Nafta disponible: 10.5 - Matricula: POO789  
Color: Rojo - Marca: Fiat  
Disponible  
////////////////////////////////////  
Estos autos no estan disponibles:  
Auto: $2230 - Nafta disponible: 11.5 - Matricula: HWP352  
Color: Verde - Marca: Nissan  
No disponible  
Auto: $3459 - Nafta disponible: 12.5 - Matricula: TRO589  
Color: Blanco - Marca: Peugeot  
No disponible  
Alquilados por los siguientes clientes  
  
Nissan) Fue entregado el dia 11.12  
Sera devuelto el dia 12.01  
Por un valor total de: 12500.0  
1-Cliente: id156  
Butassi Federico  
DNI: 41280335 - Direccion: Espania 348  
Telefono: 3571327923  
  
Peugeot) Fue entregado el dia 10.11  
Sera devuelto el dia 10.12  
Por un valor total de: 11200.0  
2-Cliente: id371  
Jeff Bezos  
DNI: 31023892 - Direccion: Dean Funes 1244  
Telefono: 3571890293
```

```

-----
Desea alquilar alguno de estos otros?

Auto: $3550 - Nafta disponible: 12.4 - Matricula: NXT345
Color: Rojo - Marca: Mazda
Disponible
Auto: $2230 - Nafta disponible: 14.2 - Matricula: PI0569
Color: Negro - Marca: Nissan
Disponible
Auto: $4539 - Nafta disponible: 16.5 - Matricula: FT0469
Color: Negro - Marca: Ferrari
Disponible
Auto: $1890 - Nafta disponible: 10.5 - Matricula: PO0789
Color: Rojo - Marca: Fiat
Disponible
Qué auto desea?
1. Mazda
2. Nissan
3. Ferrari
4. Fiat
Opción: 2
Usted seleccionó: Auto: $2230 - Nafta disponible: 14.2 - Matricula: PI0569
Color: Negro - Marca: Nissan
Disponible

Desea seleccionar otro auto (SI/NO)?:  no

Ingrese sus datos personales para confirmar la transaccion:
Nombre:
Valentina
Apellido:
Gervaldo
DNI:
41959729
Direccion:
Dean Funes 337
Telefono:
-----TRANSACCION CONFIRMADA-----
Enhorabuena Valentina Gervaldo
HA RESERVADO EL AUTO:
2 Auto: $2230 - Nafta disponible: 14.2 - Matricula: PI0569
Color: Negro - Marca: Nissan
Disponible
-----
BUILD SUCCESS
-----
Total time: 54.058 s
Finished at: 2023-07-06T23:20:44-03:00
-----

```