

Control 1 Prueba 1 Paralelo 1

Competencia a evaluar: El estudiante debe ser capaz de construir el modelo del dominio, contratos, diagrama de clases y código Java asociado, siguiendo arquitectura propuesta

Problema

Se tiene que manejar información de una compraventa de autos semi nuevos, la cual se encuentra en un archivo. Después de procesar todo el archivo, la aplicación debe realizar lo siguiente:

1. Desplegar los datos de los autos (patentes) que posee un determinado dueño, cuyo RUT se ingresa desde pantalla
2. Desplegar el dueño (rut) de un auto, cuya patente se ingresa desde pantalla

Cada registro del archivo tiene la patente de un auto y el rut de su dueño. Un dueño puede tener varios autos a la venta en dicha compraventa, máximo 10 y un auto tiene un solo dueño

Se pide (escrito en papel):

- a. Modelo del dominio completo
- b. Contratos de la interfaz del sistema
- c. Diagrama de clases
- d. Código Java de:
 - i. Clase **Dueño**
 - ii. Clase **ListaDueños**
 - iii. Código que implementa la interfaz de la App (clase que implementa la interfaz)
 - iv. Clase App completa (la que contiene el método **main**)

Notas

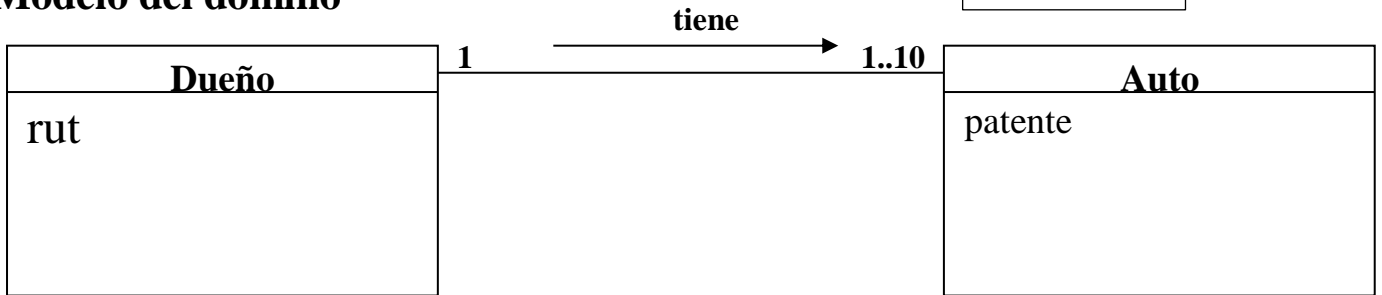
- Trabaje con excepciones para las precondiciones
- No es necesario hacer los **import**
- No es necesario trabajar con paquetes
- Para el caso de atributos de tipo primitivo, tanto en el diagrama de clases como en el código solo escriba **get... y set...**
- Si usa **toString()**, escriba el código

Recopile lo que escribió en los puntos anteriores, genere un solo PDF y suba dicho archivo PDF al trabajo en CampusVirtual.

Solución control 1

a. Modelo del domino

0.2 puntos



b. Contratos

1 punto

Operación	Ingresar auto (patente)
Descripción	Se ingresa el auto a la lista general de autos
Precondiciones	
Postcondiciones	auto ingresado a la lista general de autos

Operación	Ingresar dueño (rut)
Descripción	Se ingresa el dueño
Precondiciones	
Postcondiciones	Dueño ingresado a la lista general de dueños

Operación	Asociar dueño auto (patente, rut)
Descripción	Se ingresa el auto a la lista de autos del dueño Se asocia el auto con el dueño
Precondiciones	Que exista el dueño Que exista el auto
Postcondiciones	<ul style="list-style-type: none"> Auto ingresado a la lista de autos del dueño Auto asociado con el dueño

Operación	Obtener datos de los autos de un dueño (rut)
Descripción	Se obtiene la patente de cada uno de sus vehículos de los cuales es dueño
Precondiciones	Que exista el dueño
Post condiciones	

Operación	Obtener datos del dueño de un auto (patente)
Descripción	Se obtienen los datos del dueño del auto
Precondiciones	Que exista el auto
Post condiciones	

1 punto:

- ***0.2 cada contrato***

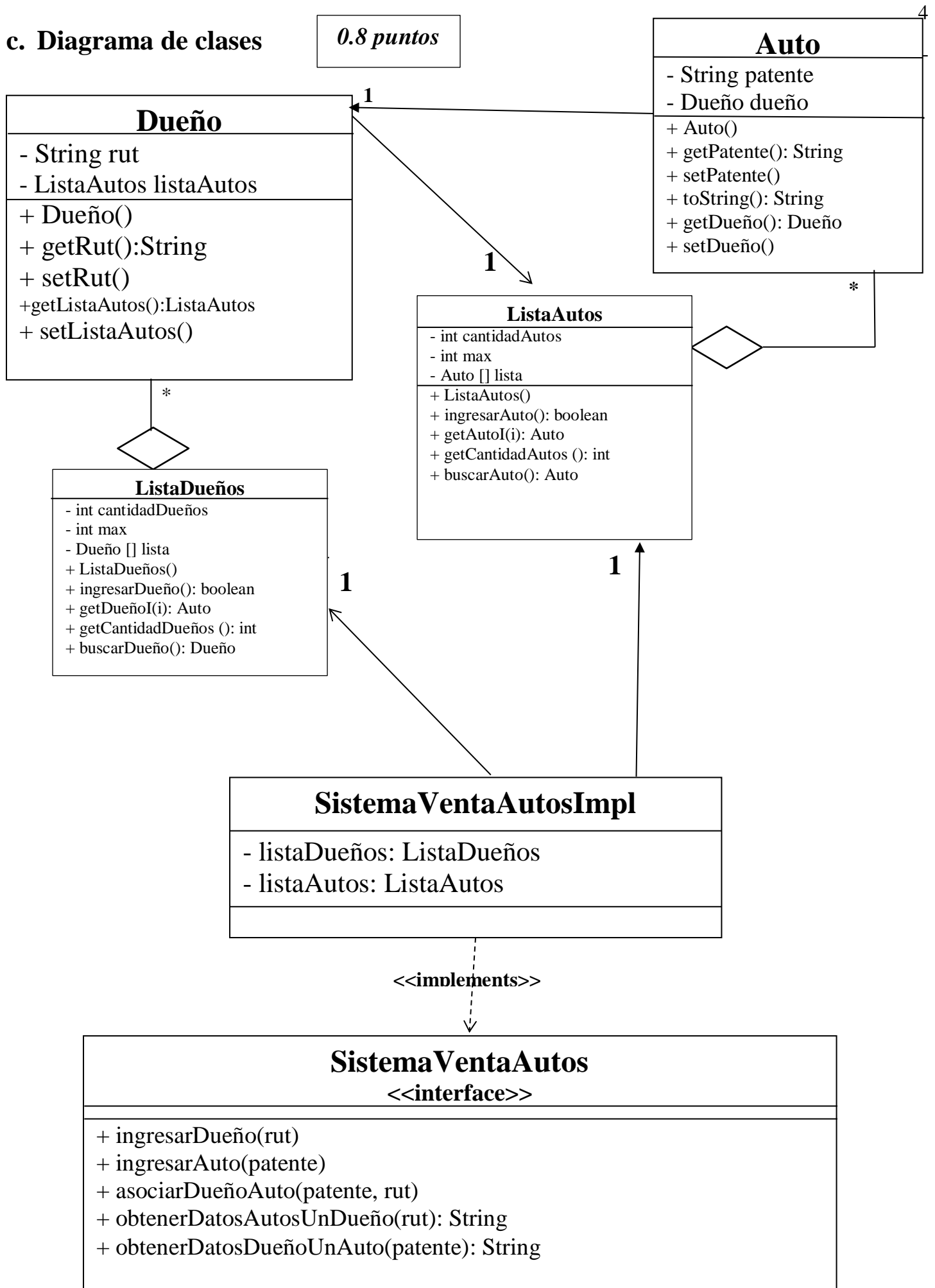
Observación

En vez de los 3 primeros contratos, se podrían tener 2 contratos:

- *Ingresar auto (patente)*
- *Ingresar dueño (patente, rut): Sería el equivalente a ingresarDueño y asociarDueñoAuto*

c. Diagrama de clases

0.8 puntos



d. Código Java**4 puntos**

```
package cl.ucn.ei.pa.sistemaCompraventa.dominio;
```

```
public class Auto {  
    private String patente;  
    private Dueño dueño;  
  
    public Auto(String patente) {  
        this.patente = patente;  
        this.dueño = null;  
    }  
  
    public String getPatente() {  
        return patente;  
    }  
  
    public void setPatente(String patente) {  
        this.patente = patente;  
    }  
  
    public Dueño getDueño() {  
        return dueño;  
    }  
  
    public void setDueño(Dueño dueño) {  
        this.dueño = dueño;  
    }  
}
```

No se pide

```
package cl.ucn.ei.pa.sistemaCompraventa.dominio;

import cl.ucn.ei.pa.sistemaCompraventa.logica.ListaAutos;

public class Dueño {
    private String rut;
    private ListaAutos listaAutos;

    public Dueño(String rut) {
        this.rut = rut;
        listaAutos = new ListaAutos(10);
    }

    public String getRut() {
        return rut;
    }

    public void setRut(String rut) {
        this.rut = rut;
    }

    public ListaAutos getListaAutos() {
        return listaAutos;
    }

    public void setListaAutos(ListaAutos listaAutos) {
        this.listaAutos = listaAutos;
    }
}
```

0.5 puntos

```

package cl.ucn.ei.pa.sistemaCompraventa.logica;

import cl.ucn.ei.pa.sistemaCompraventa.dominio.Auto;

public class ListaAutos {
    private int max;
    private int cantidadAutos;
    private Auto [] lista;

    public ListaAutos (int max) {
        lista = new Auto[max];
        cantidadAutos = 0;
        this.max = max;
    }

    public boolean ingresarAuto(Auto auto) {
        if (cantidadAutos < max) {
            lista[cantidadAutos]= auto;
            cantidadAutos++;
            return true;
        }
        else {
            return false;
        }
    }

    public Auto getAuto (int i) {
        if (i < cantidadAutos) {
            return lista[i];
        }
        else {
            return null;
        }
    }

    public Auto buscarAuto(String patente) {
        int i = 0;
        while(i < cantidadAutos && !lista[i].getPatente().equals(patente)) {
            i++;
        }
        if (i == cantidadAutos) {
            return null;
        }
        else {
            return lista[i];
        }
    }

    public int getCantidadAutos() {
        return cantidadAutos;
    }
}

```

No se pide

```
package cl.ucn.ei.pa.sistemaCompraventa.logica;

import cl.ucn.ei.pa.sistemaCompraventa.dominio.Dueño;

public class ListaDueños {
    private int max;
    private int cantidadDueños;
    private Dueño [] lista;

    public ListaDueños (int max) {
        lista = new Dueño[max];
        cantidadDueños = 0;
        this.max = max;
    }

    public boolean ingresarDueño(Dueño Dueño) {
        if (cantidadDueños < max) {
            lista[cantidadDueños]= Dueño;
            cantidadDueños++;
            return true;
        }
        else {
            return false;
        }
    }

    public Dueño buscarDueño(String rut) {
        int i = 0;
        while(i < cantidadDueños && !lista[i].getRut().equals(rut)) {
            i++;
        }
        if (i == cantidadDueños) {
            return null;
        }
        else {
            return lista[i];
        }
    }

    public Dueño getDueño (int i) {
        if (i < cantidadDueños) {
            return lista[i];
        }
        else {
            return null;
        }
    }
}
```

0.5 puntos


```
package cl.ucn.ei.pa.sistemaCompraventa.logica;
```

```
public interface SistemaVentaAutos {
    public boolean ingresarAuto(String patente);
    public boolean ingresarDueño(String rut);
    public void asociarDueñoAuto(String rut, String patente);
    public String obtenerDatosAutosUnDueño(String rut);
    public String obtenerDueñoUnAuto(String patente);
}
```

No se pide

```
package cl.ucn.ei.pa.sistemaCompraventa.logica;
```

```
import cl.ucn.ei.pa.sistemaCompraventa.dominio.Auto;
import cl.ucn.ei.pa.sistemaCompraventa.dominio.Dueño;
```

```
public class SistemaVentaAutosImpl implements SistemaVentaAutos{
    private ListaDueños listaDueños;
    private ListaAutos listaAutos;

    public SistemaVentaAutosImpl() {
        listaDueños = new ListaDueños(5);
        listaAutos = new ListaAutos(50);
    }

    public boolean ingresarAuto(String patente) {
        Auto auto = new Auto(patente);
        return listaAutos.ingresarAuto(auto);
    }

    public boolean ingresarDueño(String rut) {
        Dueño dueño = listaDueños.buscarDueño(rut);
        if (dueño == null) {
            dueño = new Dueño(rut);
            return listaDueños.ingresarDueño(dueño);
        }
        else {
            return true;
        }
    }

    public void asociarDueñoAuto(String rut, String patente) {
        Dueño dueño = listaDueños.buscarDueño(rut);
        Auto auto = listaAutos.buscarAuto(patente);
        if(dueño != null && auto != null) {
            auto.setDueño(dueño);
            dueño.getListaAutos().ingresarAuto(auto);
        }
        else {
            throw new NullPointerException("No existe auto y/o dueño");
        }
    }
}
```

2 puntos

```
public String obtenerDatosAutosUnDueño(String rut) {
    String salida ="Autos de un dueño\n";
    Dueño dueño = listaDueños.buscarDueño(rut);
    if(dueño != null) {
        for(int i=0; i<dueño.getListaAutos().getCantidadAutos(); i++){
            Auto auto = dueño.getListaAutos().getAuto(i);
            salida = salida + "patente "+ auto.getPatente()+ "\n";
        }
        return salida;
    }
    else {
        throw new NullPointerException ("No existe el dueño");
    }
}
```

```
public String obtenerDueñoUnAuto(String patente) {
    String salida ="Dueño del auto ";
    Auto auto = listaAutos.buscarAuto(patente);
    if(auto != null) {
        salida = salida + auto.getDueño().getRut();
        return salida;
    }
    else {
        throw new NullPointerException ("No existe el auto");
    }
}
```

```

package cl.ucn.ei.pa.sistemaCompraventa.logica;

import java.io.IOException;

import ucn.ArchivoEntrada;
import ucn.Registro;
import ucn.StdIn;
import ucn.Stdout;

public class App {

    public static void leerAutos(SistemaVentaAutos sistema)
        throws IOException{
        ArchivoEntrada archivo = new ArchivoEntrada("Vehiculos.txt");
        boolean ingreso = true;
        while(!archivo.isEndFile() && ingreso) {
            Registro registro = archivo.getRegistro();
            String patente = registro.getString();
            String rut = registro.getString();
            ingreso = sistema.ingresarAuto(patente);
            ingreso = sistema.ingresarDueño(rut);
            try {
                sistema.asociarDueñoAuto(rut,patente);
            }catch(NullPointerException ex) {
                StdOut.println(ex.getMessage());
            }
        }
    }

    public static void main(String[] args) throws IOException{
        SistemaVentaAutos sistema = new SistemaVentaAutosImpl();
        leerAutos(sistema);
        StdOut.print("Rut dueño: ");
        String rut = StdIn.readString();
        try {
            StdOut.println(sistema.obtenerDatosAutosUnDueño(rut));
        }catch(NullPointerException ex) {
            StdOut.println(ex.getMessage());
        }

        StdOut.print("patente: ");
        String patente = StdIn.readString();
        try {
            StdOut.println(sistema.obtenerDueñoUnAuto(patente));
        }catch(NullPointerException ex) {
            StdOut.println(ex.getMessage());
        }
    }
}

```

1 punto