

Objetivo: Que el alumno aplique los conceptos avanzados de la POO: herencia y polimorfismo, para desarrollar software flexible y de calidad, con y para reutilización, valorando las ventajas de su uso.

Nota 1: en el código manuscrito:

- no será necesario declarar ni definir los accessor's, sin embargo, deberán ser utilizados como si lo estuvieran
- los constructores deberán ser definidos en todos los casos.

Nota 2: en todos los casos, aunque no esté explícito, proveer los métodos para agregar y quitar elementos de la colección

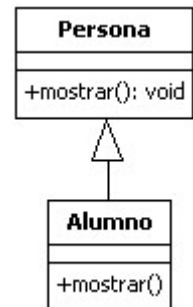
1. Realizar los cambios necesarios para reutilizar en la clase Alumno (TP2) las características generales de la clase Persona (TP3), que son heredadas, dado que un Alumno *es una* Persona, es decir, se cumple la relación de subclasificación. La relación se observa en el diagrama adjunto. El método mostrar() de la clase Persona imprime en pantalla las características de una persona, con el siguiente formato:

Nombre y Apellido: **Maria Lopez**
DNI: **25365902** Edad: **41 años**

El método mostrar() de Alumno debe redefinir el de la superclase, de modo que además de los datos personales, muestre la LU, las notas, el promedio y la leyenda correspondiente según el siguiente formato:

Nombre y Apellido: **Luis Gomez**
DNI: **28325402** Edad: **19 años**
LU: **2345** Notas: **5.0 - 9.0**
Promedio: **7.0 - DESAPROBADO**

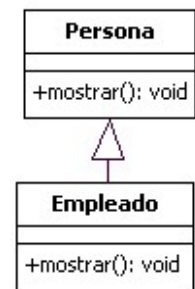
Definir una clase ejecutable Escuela, instanciar personas y alumnos y verificar el funcionamiento del método mostrar en ambos tipos de objetos.



2. Adaptar la clase Empleado (TP3) extendiendo a la clase Persona, según se observa en el diagrama de clases. Luego redefinir el método mostrar() de la superclase, de modo que además de los datos personales, muestre las características particulares de cada empleado según el siguiente diseño:

Nombre y Apellido: **Dario Perez**
DNI: **13549867** Edad: **26 años**
CUIL: **27135498678** Antigüedad: **11** años de servicio
Sueldo Neto: \$ **1236.0**

Definir una clase ejecutable Empresa, instanciar personas y empleados y verificar el funcionamiento del método mostrar en ambos tipos de objetos.



3. Implementar la clase Cuadrado según se observa en el diagrama de clases, teniendo en cuenta la clase Rectangulo definida en una serie anterior. En este caso será necesario realizar una **especialización**

Es necesario agregar el método nombreFigura(), que retorna en cada caso:

***** Rectangulo *****

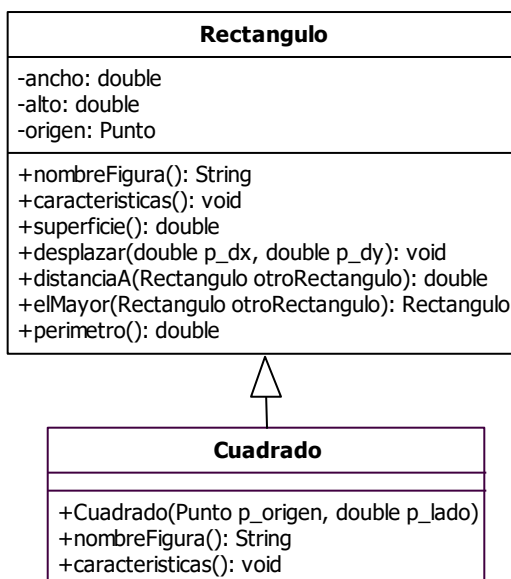
***** Cuadrado *****

El método características() del cuadrado genera el siguiente resultado:

***** Cuadrado *****

Origen: (5.0, 3.0) - Lado: 72.99

Superficie: 5327.54 - Perimetro: 291.96



4. La fórmula $\pi * r * s$ permite determinar la superficie de una elipse, donde r representa el semieje mayor y s al semieje menor, mientras que la superficie de un círculo está dada por $\pi * r * r$, o lo que es lo mismo $\pi * r^2$. Es decir que en el círculo r y s asumen el mismo valor, comúnmente llamado radio. Por lo tanto el círculo es un caso especial de elipse.

Dado el siguiente diagrama de clases, implementar la solución teniendo en cuenta las clases Punto y Círculo definidas en las series anteriores. Tanto el círculo como la elipse deben poder desplazarse desde el punto actual a otra posición. Nótese que es necesario realizar una **generalización**.

Es necesario además agregar el método nombreFigura(), que retorna en cada caso:

***** Elipse *****

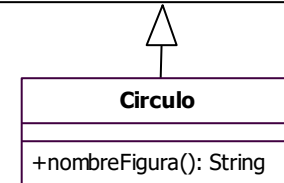
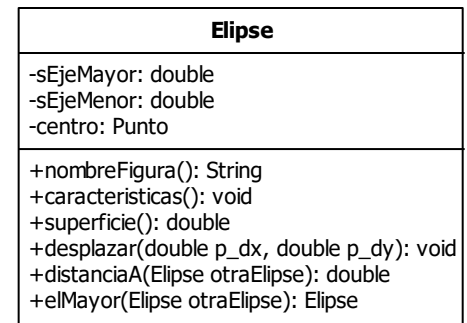
***** Círculo *****

El método características() para la elipse tiene el siguiente formato:

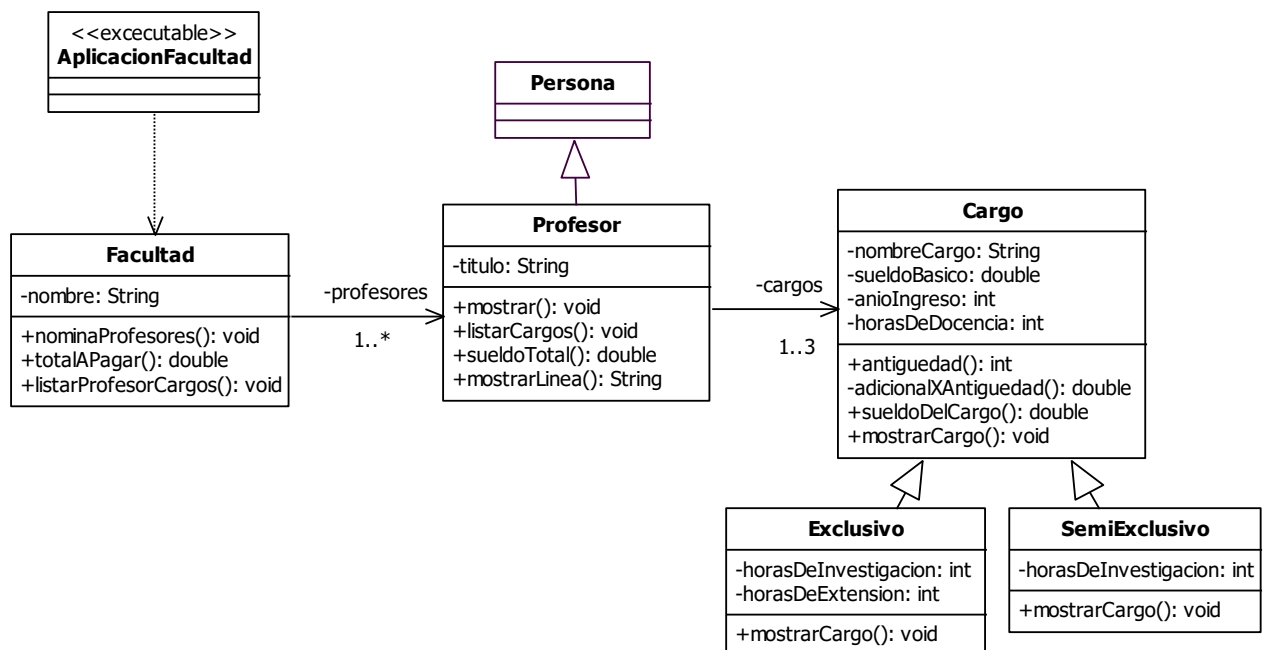
***** Elipse *****

Centro: (5.0, 3.0) - Semieje Mayor: 20.44 - Semieje Menor: 46.86

Superficie: 3009.08



5. Una facultad desea administrar información referida a los sueldos de los profesores. Para cumplir este objetivo se modela el siguiente diagrama de clases, en el que se reutilizará la declaración de la clase Persona.



Se debe tener en cuenta que un profesor no puede tener más de tres cargos. Se entiende por cargo a un puesto particular en una materia (por ejemplo JTP-Programación OO). Al momento de ser creado, el profesor ya cuenta con todos sus cargos declarados (colección).

En la clase **Cargo**, es necesario determinar la **antigüedad** de permanencia en el cargo, un **adicional** que calcule un 1% del sueldo básico por año de antigüedad, y el **sueldo** del cargo, que es la **suma del sueldo básico mas el adicional**.

Asimismo, un cargo puede ser **Simple**, **SemiExclusivo** o **Exclusivo**. Cuando el cargo es simple, la cantidad total de horas es 10, y son todas dedicadas a la docencia. En el cargo **SemiExclusivo** la cantidad total de horas es 20, con 10 hs. de docencia como mínimo, y las 10 restantes pueden ser dedicadas a la investigación o a la docencia. En el caso de los cargos **Exclusivos**, la cantidad de horas de docencia, de investigación y de extensión pueden variar.

El método **mostrarCargo()** de la clase Cargo visualiza los datos del cargo con el siguiente diseño:

JTP-Programación OO - Sueldo Basico: 800.0 - Sueldo Cargo: 904.0 – Antigüedad: 10 años
Horas Docencia: 10

En el caso de un cargo **SemiExclusivo** este método se redefine para mostrar **además**:

```
----Cargo de caracter SemiExclusivo----
Horas investigación: 10
```

En el caso de un cargo **Exclusivo** se redefine para mostrar **además**:

```
----Cargo de caracter Exclusivo----
Horas investigación: 10
Horas extension: 30
```

En la clase **Profesor** se debe obtener el sueldo total, sumando los sueldos de cada cargo que posea el profesor. Además, se desea listar los cargos de cada profesor, según el siguiente diseño:

```
-***** Cargos Asignados *****-
-----
JTP-Programación OO - Sueldo Basico: 800.0 - Sueldo Cargo: 904.0 - Antigüedad: 10 años
Horas Docencia: 10

Titular-TFA - Sueldo Basico: 1400.0 - Sueldo Cargo: 1484.0 - Antigüedad: 12 años
Horas Docencia: 20
----Cargo de caracter SemiExclusivo----
Horas investigación: 0

JTP-Analisis de Sistemas - Sueldo Basico: 800.0 - Sueldo Cargo: 872.0 - Antigüedad: 20 años
Horas Docencia: 10
```

En esta clase, el método **mostrar()** debe redefinir el de Persona, para que además de imprimir los datos básicos, agregue el título del profesor, y una lista con los datos de sus cargos y el monto total a cobrar, como se observa en el ejemplo siguiente:

```
Nombre y Apellido: Juan Perez
DNI: 28596132 Edad: 37 años
Titulo: Lic. en Sistemas de Información

-***** Cargos Asignados *****-
-----
JTP-Programación OO - Sueldo Basico: 800.0 - Sueldo Cargo: 904.0 - Antigüedad: 10 años
Horas Docencia: 10

Titular-TFA - Sueldo Basico: 1400.0 - Sueldo Cargo: 1484.0 - Antigüedad: 12 años
Horas Docencia: 20
----Cargo de caracter SemiExclusivo----
Horas investigación: 0

JTP-Analisis de Sistemas - Sueldo Basico: 800.0 - Sueldo Cargo: 872.0 - Antigüedad: 20 años
Horas Docencia: 10

** Sueldo Total: 3260.0 **
```

Además la clase **Profesor** provee una cadena con el siguiente formato, en el método **mostrarLinea()**

```
DNI: 28.596.132 - Nombre: Juan Perez - Sueldo Total: 3260.0
```

La clase **Facultad** es la responsable de emitir una nómina de profesores, que incluye el total a pagar, con el siguiente diseño:

```
***** Nómina Facultad: FaCENA
-----
DNI: 21.859.361 Nombre: Juan Perez - Sueldo Total: 3260.0
DNI: 23.451.992 Nombre: Mirta Lopez - Sueldo Total: 3306.0
-----
Total a pagar: $6566.0
```

Esta clase también es responsable de proveer un listado de los cargos de todos los profesores, con el siguiente diseño:

```
***** Detalle de Profesores y cargos de Facultad: FaCENA *****
```

Nombre y Apellido: Juan Perez
DNI: 28596132 Edad: 37 años
Titulo: Lic. en Sistemas de Información

-***** Cargos Asignados *****-

JTP-Programación OO - Sueldo Basico: 800.0 - Sueldo Cargo: 904.0 - Antigüedad: 10 años
Horas Docencia: 10

Titular-TFA - Sueldo Basico: 1400.0 - Sueldo Cargo: 1484.0 - Antigüedad: 12 años
Horas Docencia: 20
----Cargo de caracter SemiExclusivo----
Horas investigación: 0

JTP-Analisis de Sistemas - Sueldo Basico: 800.0 - Sueldo Cargo: 872.0 - Antigüedad: 20 años
Horas Docencia: 10

** Sueldo Total: 3260.0 **

----- *** -----

Nombre y Apellido: Mirta Lopez
DNI: 34519921 Edad: 30 años
Titulo: Lic. en Sistemas de Información

-***** Cargos Asignados *****-

JTP-TP-II - Sueldo Basico: 1200.0 - Sueldo Cargo: 1344.0 - Antigüedad: 11 años
Horas Docencia: 10

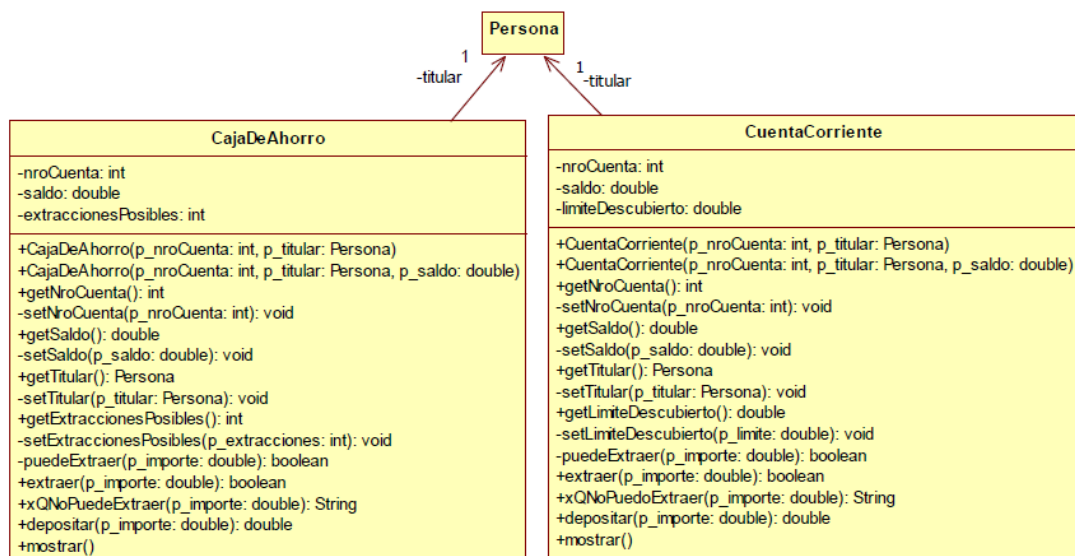
JTP-Ing.Soft - Sueldo Basico: 1800.0 - Sueldo Cargo: 1962.0 - Antigüedad: 10 años
Horas Docencia: 10
----Cargo de caracter Exclusivo----
Horas investigación: 20
Horas extension: 10

** Sueldo Total: 33060.0 **

Hay 2 profesores

Definir una clase ejecutable AplicacionFacultad y emitir un listado de los profesores con sus cargos. Por otra parte, el Departamento Contable desea una nómina de los profesores, para realizar el pago del corriente mes.

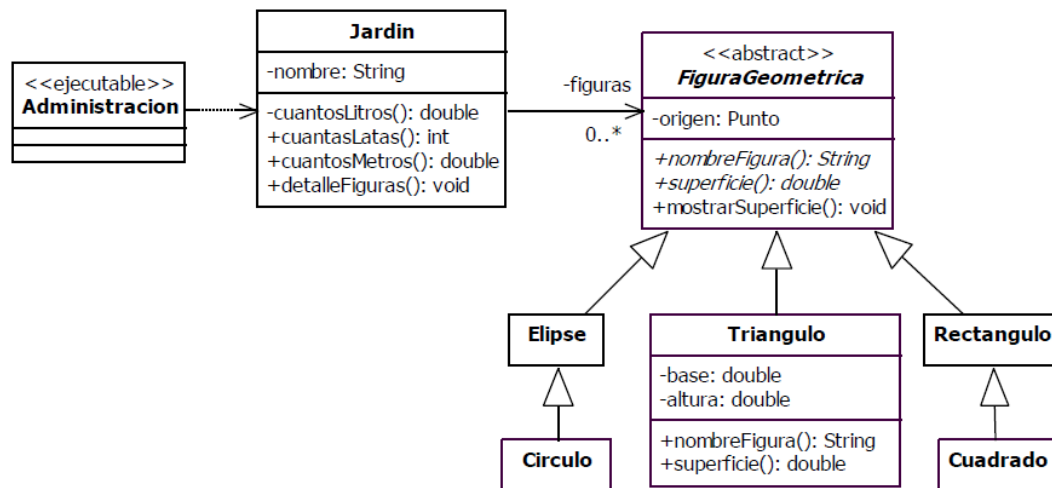
6. El diagrama de clases adjunto, implementado anteriormente (TP 3), presenta las características de las cuentas de los clientes de un banco. Se observan en ellas características similares, que pueden ser generalizadas y luego reutilizadas.



Realice en primer lugar el análisis del problema y diseñe el diagrama de clases apropiado, generalizando las características comunes. Luego modifique el código de modo que se adecúe al nuevo diseño.

ATENCIÓN ¡! Constatar que la clase Banco implementada anteriormente no sufre ningún cambio, y puede ser ejecutada sin inconvenientes.

7. Un jardín de niños desea pintar figuras en el piso, y necesita saber la cantidad de pintura que debe comprar, en función de la superficie a pintar. Si se conoce que una lata de 4 litros cubre 20 m², implementar en una clase Jardín los cálculos necesarios para determinar la cantidad de latas a comprar. El diseño que representa la solución se muestra en el siguiente diagrama de clases. Reutilizar las clases Punto, Circulo, Elipse, Rectángulo y Cuadrado definidas con anterioridad. Definir la clase Triangulo, cuya estructura está compuesta de una base y una altura, más un punto de origen, que indica el extremo inferior izquierdo. Implementarla sólo con los métodos nombreFigura() y superficie().



Para cualquier figura geometrica, el método mostrarSuperficie() es responsable de visualizar los resultados de calcular la superficie de cualquier figura, según el siguiente diseño:

```

***** Cuadrado *****
Superficie: 2.25

***** Triángulo *****
Superficie: 1.28

***** Círculo *****
Superficie: 10.17

***** Elipse *****
Superficie: 47.12
    
```

El detalle de figuras debe emitir un listado de todas las figuras que pintará el jardín, y que incluya el total de superficie a cubrir y la cantidad de latas que se necesita comprar. El detalle impreso tiene el siguiente diseño:

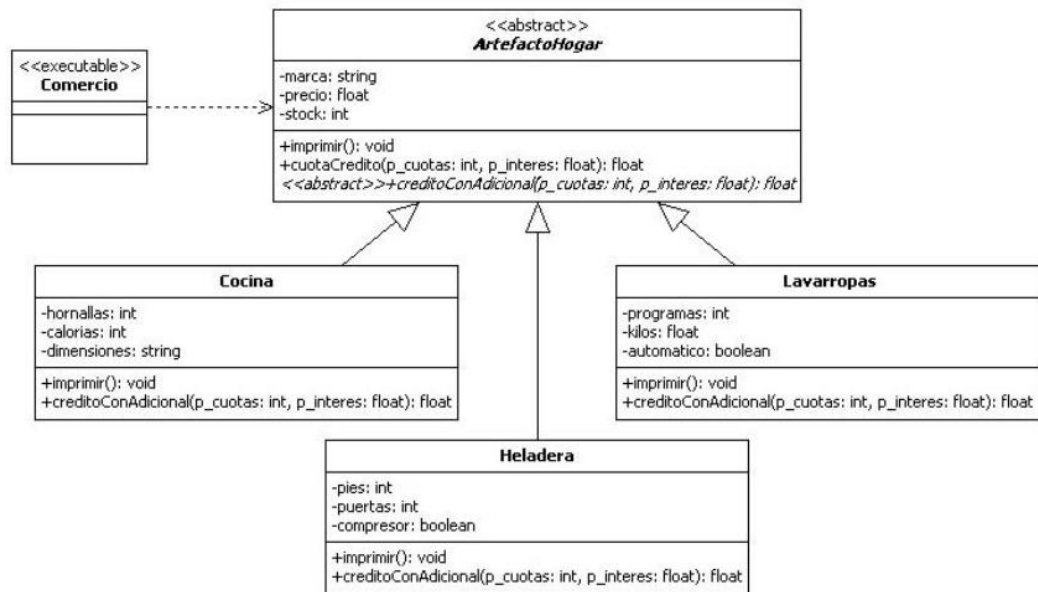
```

Presupuesto: Bichito de Luz

***** Cuadrado *****
Superficie: 25.0
***** Triángulo *****
Superficie: 1.28
***** Círculo *****
Superficie: 12.57
***** Elipse *****
Superficie: 47.12
-----
Total a cubrir: 85.97
Comprar 5 latas
    
```

En la clase ejecutable Administración, instanciar distintas figuras, simulando lo que desea pintar el jardín, y emitir un detalle con la cantidad de latas a comprar.

8. Un comercio de venta de electrodomésticos desea implementar una aplicación que le permita simular la compra en cuotas, para informar al cliente en el momento previo a la adquisición. El siguiente diagrama de clases representa el diseño de la situación:



El cálculo del valor de la cuota al solicitar un crédito es común a todos los artefactos: al precio original se agrega el interés y se divide por el número de cuotas pactadas.

En la superclase el método imprimir muestra los datos comunes a todos los artefactos

Marca: Volcan - Precio: 600.0 - Stock: 15

Según el artefacto, el valor de la cuota podrá tener un cálculo adicional: una heladera, si posee compresor, pagará \$50 adicionales por cada cuota en concepto de un seguro. En el caso de un lavarropas, si no es automático, se realizará un descuento de 2% en cada cuota.

En cada artefacto concreto, el método imprimir, además de los datos comunes a todos los artefactos, debe imprimir los datos particulares. Por ejemplo:

```

**** Cocina ****
Marca: Volcan - Precio: 600.0 - Stock: 15
Hornallas: 4
Calorias: 2000
Dimensiones: 80*60*60cm
  
```

En la clase ejecutable Comercio, crear varios artefactos, que serán los que ofrece el comercio. Luego simular consultas de clientes, para diferentes situaciones, variando la cantidad de cuotas y el interés aplicado. La salida debe proveer la siguiente información:

```

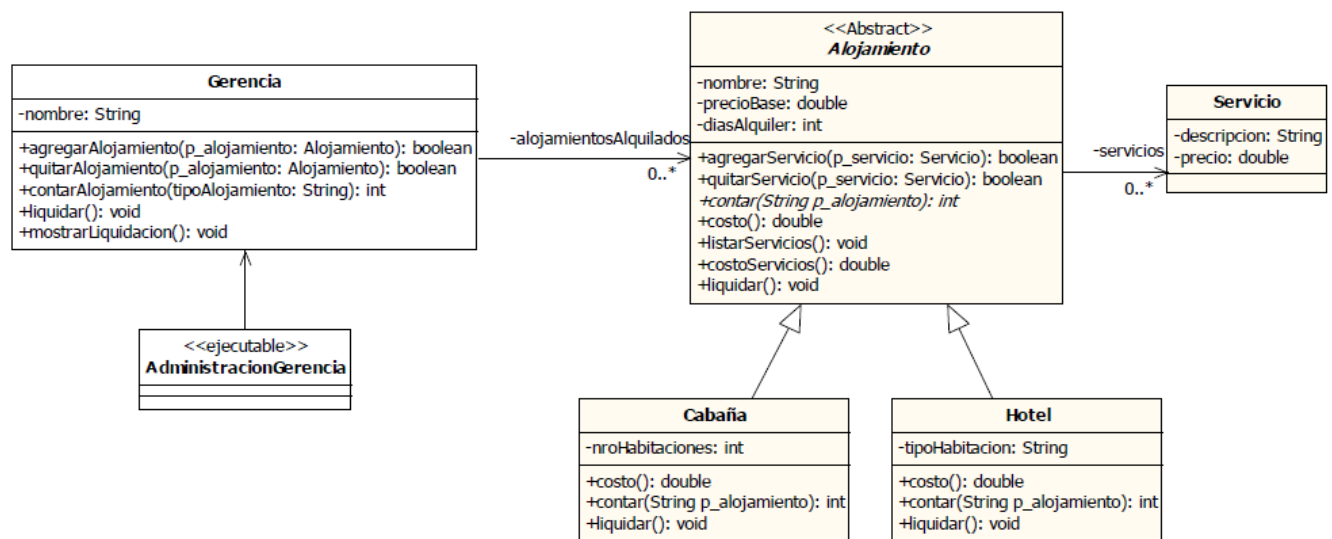
**** Cocina ****
Marca: Volcan - Precio: 600.0 - Stock: 15
Hornallas: 4
Calorias: 2000
Dimensiones: 80*60*60cm
Cuotas: 6 - Interes: 2.53%
Valor Cuota: 115.18
Valor Cuota Con Adicionales: 115.18

**** Heladera ****
Marca: Gafa - Precio: 1200.0 - Stock: 8
Pies: 11
Puertas: 2
Compresor: Si
Cuotas: 6 - Interés: 2.53%
Valor Cuota: 230.36
Valor Cuota Con Adicionales: 280.36
  
```

9. Una cadena hotelera desea automatizar parte de su información con el fin de agilizar la liquidación de la estadía en sus alojamientos. La información registrada se refiere a los tipos de alojamiento que maneja (cabaña/ hotel), y los servicios extras asociados al alojamiento (alquiler de automóvil, lavandería, etc). Toda la información es registrada por una

gerencia regional. En cada alojamiento el precio base es el precio a abonar por cada día, sin considerar los servicios. Al momento de liquidar la estadía, se deberá tener en cuenta la cantidad de días y las características particulares de cada tipo de alojamiento.

El diseño de la solución se encuentra en el diagrama de clases adjunto.



La aplicación desarrollada debe permitir registrar o anular, en relación al alojamiento, un servicio solicitado por el cliente, y proveer un listado de los servicios utilizados efectivamente (Ej: alquiler auto: \$100.0)

El costo del alojamiento se calcula en función del precio base y la cantidad de días de alquiler, mientras que el costo de los servicios será la suma de todos los servicios utilizados por el cliente.

La liquidación de la estadía, común a los distintos tipos de alojamiento, incluidos los servicios, tiene el formato siguiente:

```

Alojamiento: Cabañas La Alondra
Costo por 5 días: $600.0
alquiler auto: $100.0
lavanderia: $20.0
  
```

En el caso particular de las **cabañas**, al costo calculado para el alojamiento se debe agregar \$30 por día por cada habitación que tenga la cabaña. Al liquidar el alquiler de la cabaña, se debe agregar a la liquidación la característica de la cabaña y el costo total. El formato es el siguiente:

```

Cabaña con 3 habitaciones
Total: -----> $1170.0
  
```

Al calcular el costo de los **hoteles**, al costo base se debe agregar \$20 si el tipo de habitación es “Single” o \$35 si es habitación “Doble”, por la cantidad de días alojado. Al liquidar el alquiler del hotel, se debe agregar a la liquidación la característica de la habitación del hotel y el costo total. El formato es el siguiente:

```

Habitacion Single
Total: -----> $735.0
  
```

Con frecuencia la **gerencia** desea saber la cantidad de cabañas y de habitaciones de hotel que tiene alquiladas. Por tal motivo, es responsabilidad de cada clase concreta (Hotel o Cabaña), permitir contar un alojamiento más, en cada tipo de alojamiento particular, de manera que al recorrer la colección de alojamientos alquilados, se pueda contar la cantidad de alquileres de un tipo determinado (Hotel o Cabaña), que se ingresa como parámetro.

Por su parte la gerencia tiene la responsabilidad de llevar un registro de los alojamientos que están alquilados.

Debe implementar también el método apropiado para contar un tipo de alojamiento determinado, pudiendo ser “Hotel” o “Cabaña”. Debe liquidar la estadía para cada hotel o cabaña alquilado, y mostrar la liquidación, que será un resumen con el formato siguiente:

```

Gerencia Los Arroyos
Liquidación-----

Alojamiento: Cabañas La Alondra
Costo por 5 días: $1050.0
    alquiler auto: $100.0
  
```



```

        lavanderia: $20.0
        Cabaña con 3 habitaciones
Total: -----> $1170.0
Alojamiento: Hotel Guaraní
Costo por 7 días: $630.0
        internet: $5.0
        alquiler auto: $100.0
        Habitación Single
Total: -----> $735.0

Alojamiento tipo Cabaña ---->1
Alojamiento tipo Hotel ----->1
    
```

Implementar una clase ejecutable Administración Gerencia, que realice la liquidación para las siguientes situaciones:

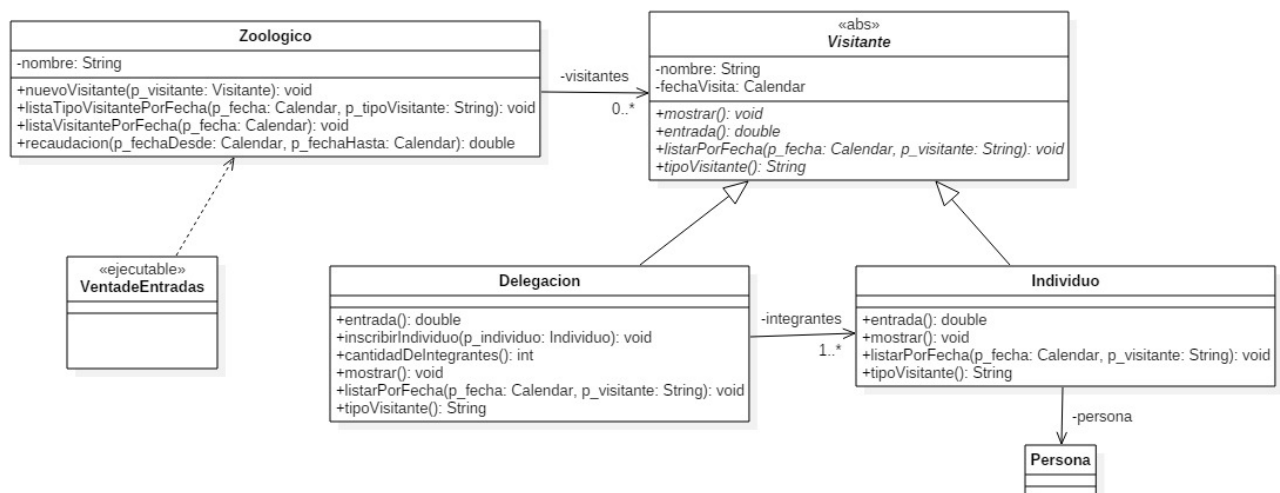
- Una habitación de hotel, con una estadía de 1 semana, con servicio de internet y lavandería
- Una cabaña, por 5 días, con servicio de lavandería y alquiler de auto

10. Un zoológico desea llevar registro de las entradas que vende. El valor de la entrada depende del tipo de visitante, que puede ser: **individuo** ó **delegación**. La clase **Persona** colabora con **Individuo** para prestarle sus servicios. De cada visitante se registra **nombre y fecha** de la visita. Una delegación tiene un nombre que la identifica y está formada por una **colección de individuos**. La entrada por individuo es de \$10. El costo de la entrada de una delegación es la suma de las entradas de sus integrantes, menos un 10% de descuento.

El zoológico desea conocer:

- El listado de visitantes que asistieron un determinado día.
- El listado de un tipo de visitante (Individuo/Delegacion) que asistieron un determinado día.
- La recaudación total en un rango de fechas.

El diseño de la solución está plasmado en el siguiente diagrama de clases:



Nota: reutilizar la clase Persona, definida en una serie de trabajos prácticos anterior.

Para determinar el tipo de visitante, cada clase concreta retornará una cadena de caracteres que indicará el tipo en cada caso ("Individuo" ó "Delegacion")

Ambos tipos de visitantes deben implementar un listado, en una fecha determinada, para cada tipo de visitante en particular. Es decir, Individuo permitirá listar los individuos, y Delegacion permitirá listar las delegaciones. El formato de impresión dependerá del tipo de visitante. En el caso de un solo individuo será:

```

Nombre y Apellido: María Perez
DNI: 14528796 Edad: 56 años
    
```

Para las delegaciones el formato será:

```

-----
Delegación: UNNE - Alumnos 2do. Año
    
```

```

Integrantes
Nombre y Apellido: María Louseau
DNI: 13568963 Edad: 56 años
Nombre y Apellido: Carlos Lopez
    
```


DNI: 19875785 Edad: 47 años

...

Cantidad de Integrantes: 15

La clase **Delegacion** tiene además la responsabilidad de inscribir un individuo a la delegación, y de conocer la cantidad de individuos que son integrantes de la delegación.

La clase **Zoologico** por su parte tiene la responsabilidad de agregar un visitante, calcular la recaudación en un rango de fechas determinado, y proveer listados de visitantes en una fecha en particular. Debe permitir listar todos los visitantes sin discriminar, o un tipo en particular (“Delegación” o “Individuo”).

Se solicita implementar una **clase ejecutable** para realizar lo siguiente:

- Instanciar un zoológico con el nombre “El Caribú”
- Instanciar **tres individuos** que asistieron al zoológico, uno de ellos en forma independiente y los otros dos formando parte de la delegación “PAMI”, el día 22/09/2016
- Crear la delegación “PAMI”, que asistió el día indicado, e inscribir a los dos individuos.
- Listar todos los visitantes que acudieron al zoológico el día 22/09/2016.
- Listar la recaudación del último mes.
- Listar las delegaciones que acudieron el día 22/09/2016

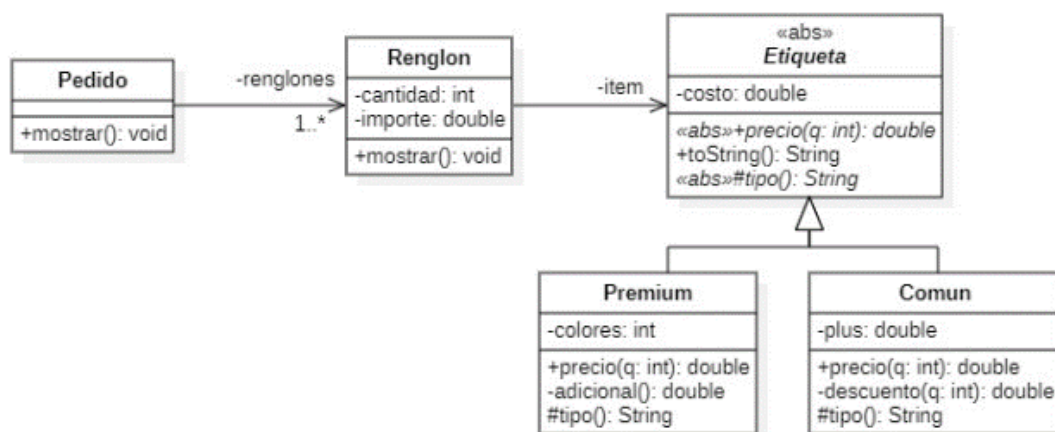
11. La empresa de diseño “Etiquet.AR” ofrece dos tipos de etiquetas para packaging: comunes y premium. Las etiquetas comunes tienen hoy un costo de \$100 por unidad. Para obtener el **precio de venta se** debe sumar el costo de las unidades vendidas y se debe adicionar un **plus por el diseño**. Por otra parte, puede tener un descuento por cantidad de unidades, según el siguiente cuadro:

Cantidad de unidades	Descuento por cantidad
1 a 10	0%
11 a 50	2%
51 a 100	5%
+100	1% cada 10u

Para las etiquetas **premium** existe un costo base acordado con el cliente al momento de hacer el pedido. A este costo se le agrega un adicional de acuerdo con los colores que se utilicen según la siguiente tabla

Cantidad de colores	Adicional por color
1	0%
2	5%
3	7%
+3	3% cada color

El diseño de la solución se observa en el siguiente diagrama de clases:



- La clase **Pedido** muestra:

Pedido:

Cantidad de items: 3

Item 1: 7 Etiquetas de tipo Premium - Costo: \$200.0 - Colores: 10

Precio: \$ 1820.0

Item 2: 57 Etiquetas de tipo Común - Costo: \$100.0 - Diseño: \$100.0

Precio: \$ 5510.0

Item 3: 94 Etiquetas de tipo Premiun - Costo: \$200.0 - Colores: 7

Precio: \$ 22748.0

--- Total pedido: 158 Etiquetas por un importe total de: \$ 30078.0

- La clase **Renglon** muestra:
7 Etiquetas de tipo Premiun - Costo: \$200.0 - Colores: 10
- En la clase **Etiqueta** el método toString() devuelve una cadena con el formato:
tipo Premiun - Costo: \$200.0 - Colores: 10
o con el formato:
tipo Común - Costo: \$100.0 - Diseño: \$100.0
Según sea una instancia de la clase Premium o de la clase Comun.

NOTA: en negritas el texto que no cambia.

Se solicita:

- a) Implementar todas las clases del diagrama con sus constructores. Codificar sólo los accesors de las clases Pedido y Etiqueta. Usar todos los accesors como si estuvieran implementados.
- b) Crear una clase ejecutable que permita simular el armado de un pedido, introduciendo valores constantes, que muestre por pantalla el detalle de la salida tal cual se muestra en el ejemplo anterior.