

# Programación Avanzada

## Examen Diciembre 2012

**Por favor siga las siguientes indicaciones:**

- Escriba con lápiz.
- Escriba las hojas de un solo lado.
- Escriba su nombre y número de documento en todas las hojas que entregue.
- Numere las hojas e indique el total de hojas en la primera de ellas.
- Recuerde entregar su número de examen junto al examen.
- El examen tiene una duración de 3hs.

### **Problema 1 (30 puntos)**

La División de Laboratorios Veterinarios del Uruguay está interesada en contar con una aplicación de vademécum (catálogo especializado) de productos veterinarios. Para ello relevó la siguiente información:

Un laboratorio elabora distintos productos veterinarios de forma exclusiva. Para elaborar productos, el laboratorio utiliza un conjunto de componentes autorizados, identificados por su nombre, y con un tipo (principio activo o excipiente). Un producto veterinario se identifica por su nombre comercial y posee una fórmula. La fórmula indica la cantidad de cada componente presente en el producto en una unidad determinada (miligramos, gramos, mililitros o litros).

Los productos se clasifican como de consumo interno o exclusivo para exportar. En caso de ser exclusivo para exportar interesa saber los países a los que se exporta.

Existe la posibilidad de que un laboratorio firme un acuerdo comercial con otro para ser el nuevo elaborador de un producto veterinario. Se desea registrar la fecha del acuerdo comercial, el producto y los laboratorios participantes (actual y nuevo elaborador). Como parte del acuerdo se le permite al nuevo laboratorio agregar componentes excipientes a la fórmula del producto. Las condiciones del producto para realizar un acuerdo comercial son que sea para consumo interno y que no tenga otro acuerdo comercial.

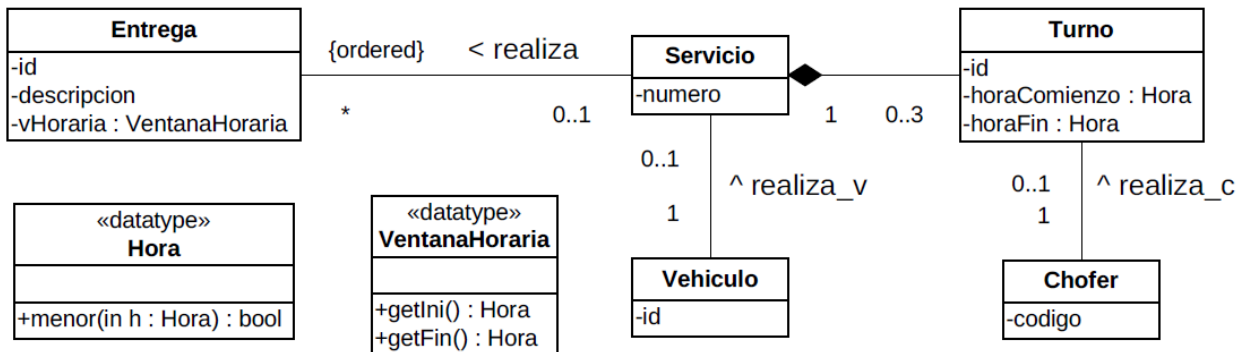
Se requiere que la aplicación a desarrollar permita realizar búsquedas de productos veterinarios por nombre de laboratorio elaborador, nombre de país al que se exporta y nombre de un componente de la fórmula del producto.

**Se pide:**

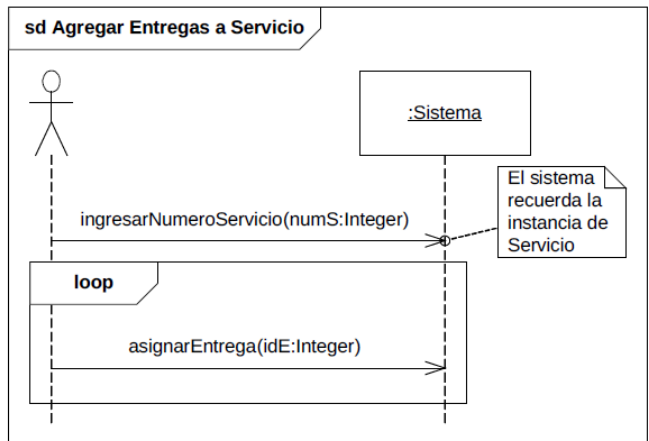
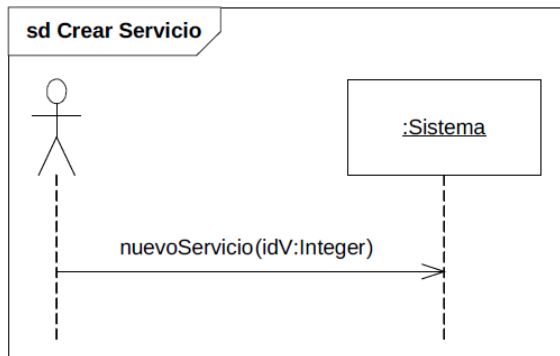
1. Modelar la realidad planteada mediante un Diagrama de Modelo de Dominio UML.
2. Expresar todas las restricciones del modelo en **lenguaje natural**.

## Problema 2 (35 puntos)

La sección distribución de una compañía de servicios, se encuentra en proceso de elaboración de un software de apoyo a la planificación de su operativa diaria. Dicha operativa consiste en realizar un conjunto de entregas, que son asignadas a un servicio (ruta que realiza un vehículo durante todo un día), que a su vez está dividido en varios turnos (cada uno asociado a un chofer que maneja el vehículo correspondiente al servicio). La figura muestra un modelo conceptual realizado para representar la realidad descrita.



Se cuenta con diagramas de secuencia del sistema de los casos de uso **Crear Servicio** y **Agregar Entregas a Servicio**, junto con los contratos parciales de las operaciones del sistema asociadas a dichos casos.



<b>Operación</b>	nuevoServicio(idV:Integer);
<b>Pre y postcondiciones</b>	
<p><b>pre:</b> Existe en el sistema un Vehiculo, cuyo atributo id tiene el valor idV.</p> <p><b>post:</b> Existe una nueva instancia de Servicio, cuyo atributo num coincide con el último número de servicio disponible en el Sistema.</p> <p><b>post:</b> Se incrementa en uno el valor del último número de servicio disponible en el Sistema.</p> <p><b>post:</b> Existe un nuevo link entre la instancia creada de Servicio y la instancia de Vehiculo identificada por idV.</p>	

<b>Operación</b>	ingresarNumeroServicio(numS:Integer);
<b>Pre y postcondiciones</b>	
<p><b>pre:</b> Existe en el sistema un Servicio, cuyo atributo numero tiene el valor numS.</p> <p><b>post:</b> Existe en la memoria del sistema un Servicio, cuyo atributo numero tiene el valor numS.</p>	

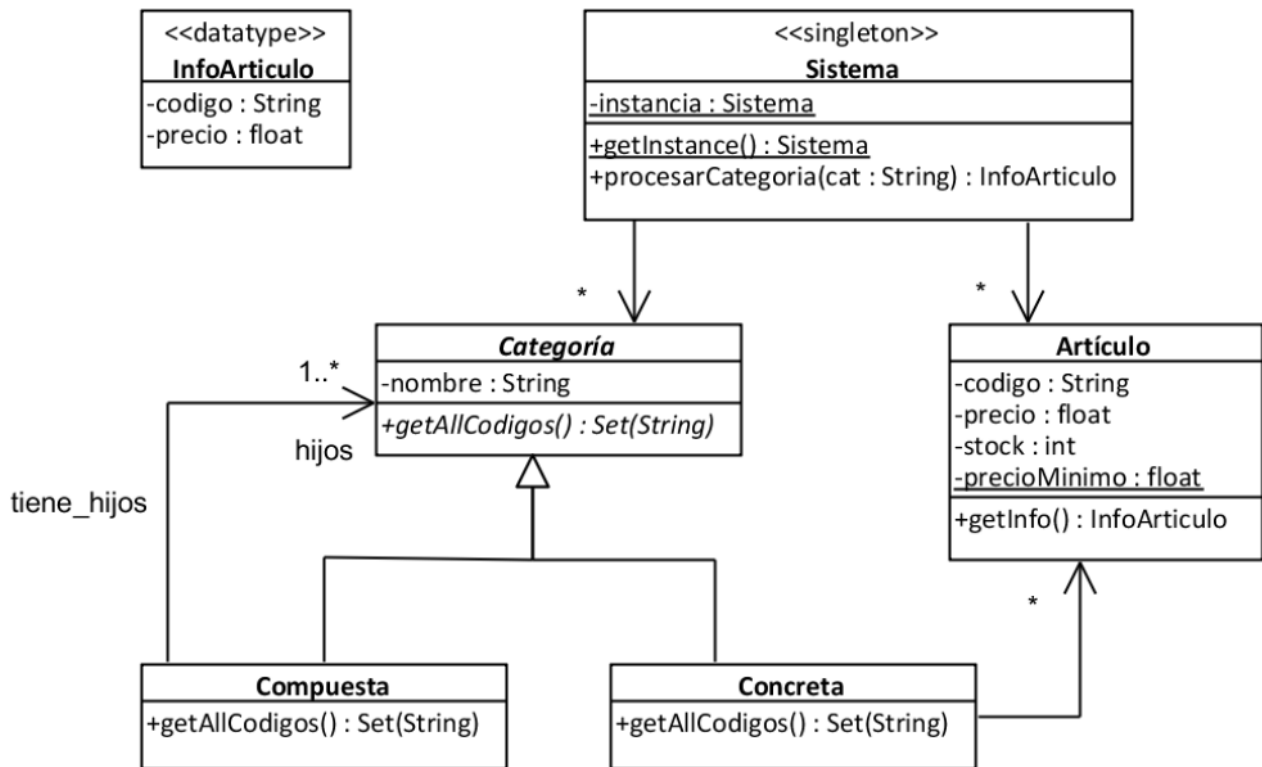
<b>Operación</b>	asignarEntrega(idE:Integer);
<b>Pre y postcondiciones</b>	
<p><b>pre:</b> Existe en el sistema una Entrega, cuyo atributo id tiene el valor idE.</p> <p><b>pre:</b> Existe en la memoria del sistema un Servicio (denominado serv).</p> <p><b>post:</b> Existe un nuevo link entre la Entrega cuyo atributo id tiene el valor idE y el Servicio serv.</p>	

Se pide:

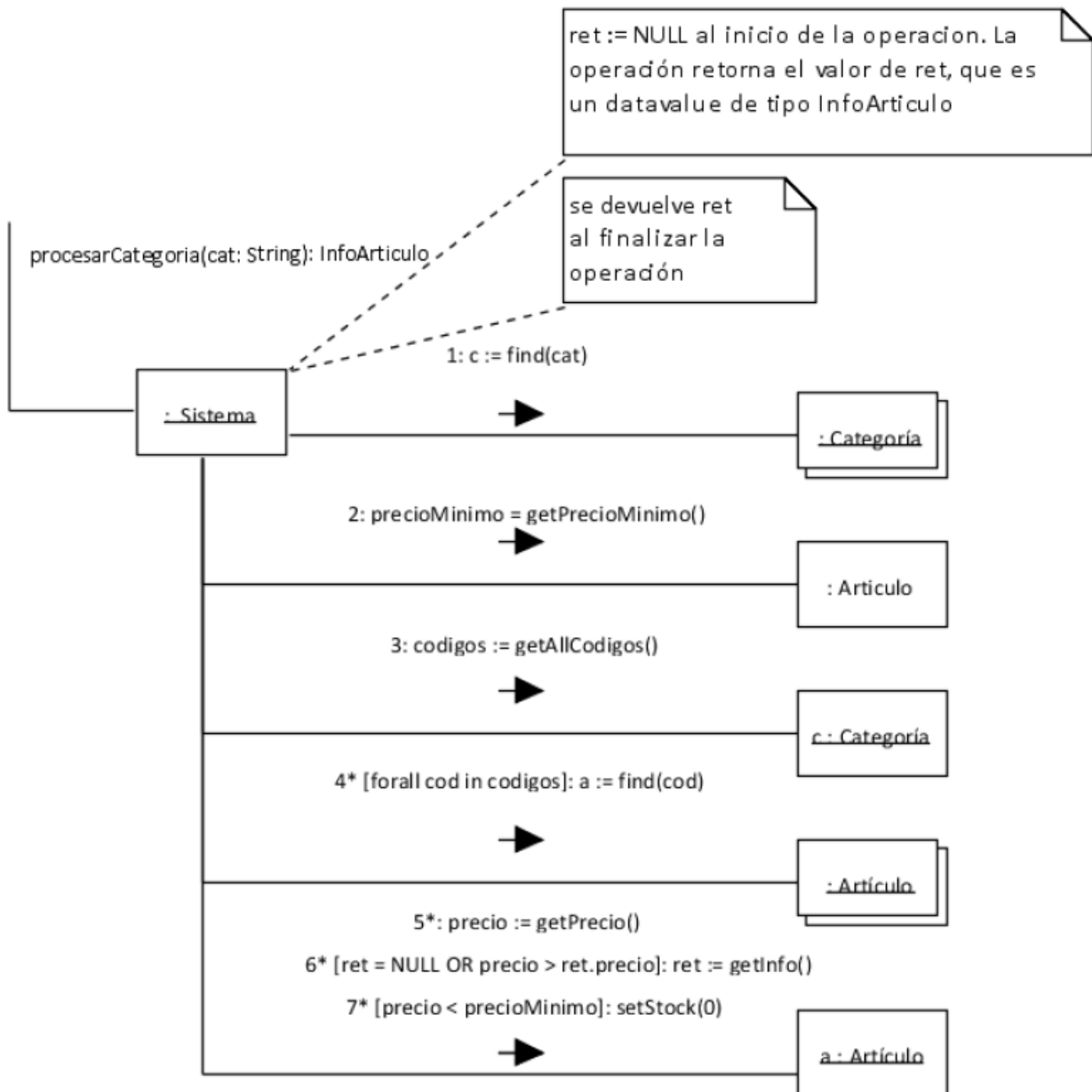
- Realizar los Diagramas de Comunicación de las 3 operaciones del sistema.
- Realizar el Diagrama de Clases de Diseño resultante.

### Problema 3 (35 puntos)

Se desea implementar un sistema que gestiona el inventario de artículos de una tienda. Los artículos tienen un código, precio y se conoce su stock. Además, los artículos se catalogan en distintas categorías que pueden estar anidadas dentro de otras categorías. A continuación se presenta un diagrama de clases de diseño parcial de la solución diseñada.

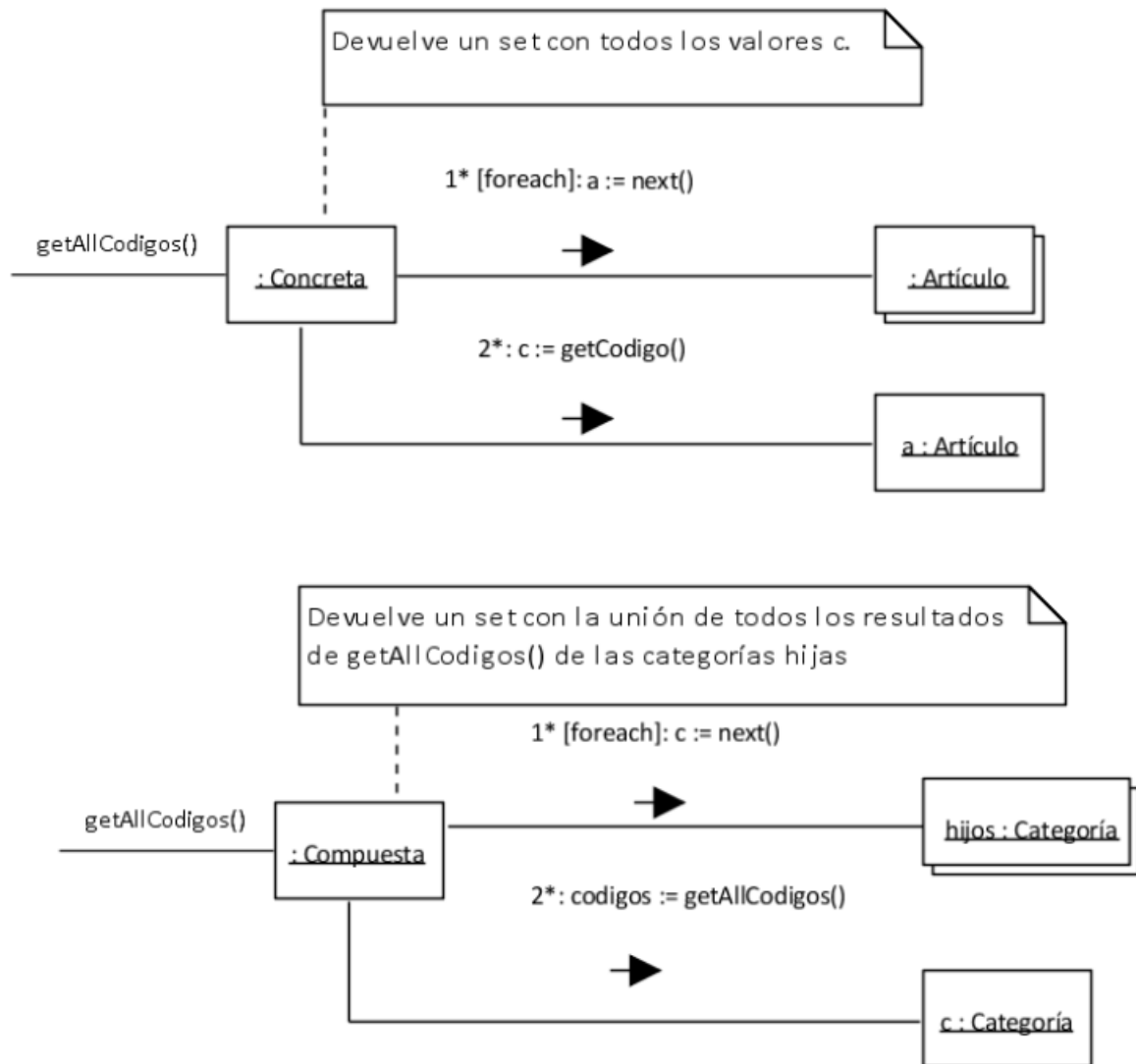


La operación `procesarCategoría(cat: String)` reduce a 0 el stock de todos los artículos de la categoría `cat` cuyo precio es menor a `Artículo::precioMínimo` y devuelve la información del artículo de precio mayor que esté asociado a dicha categoría usando la operación `Categoría::getAllCódigos()`. Se tiene el siguiente diagrama de comunicación para la operación `procesarCategoría()`.



La operación `procesarCategoria()` tiene como precondition que exista una categoría de nombre igual a `cat` en el sistema y en caso contrario lanza una excepción de tipo `std::invalid_argument`.

La operación abstracta `getAllCodigos()` devuelve un conjunto con todos los códigos de los artículos asociados a la misma categoría (si es `Concreta`) o asociados recursivamente a través de la asociación `tiene_hijos` (si es `Compuesta`). A continuación los diagramas de comunicación correspondientes a dicha operación.



**Se pide:**

- Implementar en C++ los .h de las clases Sistema, Artículo, Categoría y Compuesta.
- Implementar en C++ los .cpp de las clases Sistema y Compuesta.

**Notas:**

- Deben verificarse las precondiciones de las operaciones.
- No debe incluirse get y set de atributos.
- Suponga que se tiene una implementación de la clase **String** que implementa **ICollection** e **IKey**.
- Suponga que se tiene una implementación de **ICollection** llamada **Col** e **IDictionary** llamada **Dict**.
- No incluya constructores ni destructores de las clases.
- No incluya las directivas del preprocesador.