

Programación Avanzada

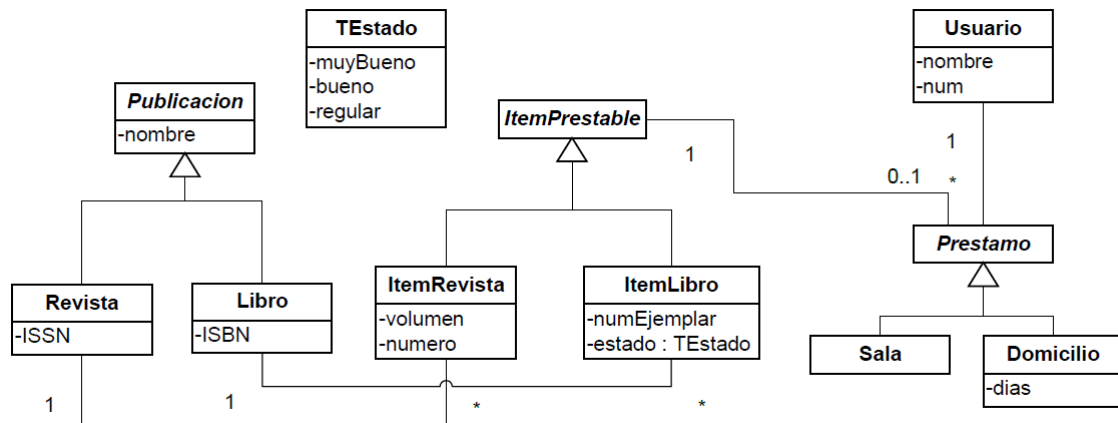
SEGUNDO PARCIAL 2014

Problema 1 (10 puntos)

- Explicar brevemente en qué consiste cada uno de los siguientes criterios GRASP: *Bajo acoplamiento*, *Alta cohesión*, *No hables con extraños*.
- ¿Qué tipos de visibilidad pueden existir entre objetos, en el contexto de un diagrama de comunicación, y cuándo se dan estas visibilidades?
- ¿Qué relación existe entre controladores e interfaces del sistema?
- Mencione brevemente cómo se implementa una interfaz en C++.
- ¿Cómo se define una colección concreta a partir de una colección genérica?

Problema 2 (25 puntos)

La biblioteca de un instituto mantiene información acerca de los libros y revistas que tiene, los usuarios que acceden a dichas publicaciones y los respectivos préstamos. La figura muestra un modelo de dominio del problema, donde las publicaciones representan la descripción de las mismas, mientras que los ítems prestables representan los volúmenes tangibles que pueden ser prestados a los usuarios. Las instancias de *ItemLibro* correspondientes a un mismo *Libro* se diferencian por el valor de su atributo *numEjemplar* (identificador numérico único autogenerated por el sistema); el atributo *estado* indica el deterioro del ejemplar debido a su uso. Las instancias de *ItemRevista* se identifican por su *volumen* y *numero* (asignados por la editorial de la *Revista* correspondiente); la biblioteca tiene un solo ejemplar de cada (*volumen*, *numero*). Los préstamos pueden ser para *Sala* o *Domicilio*; en el último caso se registra por cuantos días se otorgó. Se desea registrar solamente los préstamos actuales, no el historial de los mismos.



Se consideran los casos de uso *Ingresar nuevo libro* y *Obtener préstamos de un libro*, cada uno de los cuales es modelado con una sola operación del sistema, cuyos contratos se especifican a continuación.

ingrNuevoLibro(nl, isbn : String; es: Set(TEstado))	
Descripción	Da de alta un nuevo libro en el sistema junto con sus ejemplares.
Precondiciones	- No existe en el sistema un <i>Libro</i> de nombre <i>nl</i> .
Postcondiciones	<ul style="list-style-type: none"> - Existe en el sistema una nueva instancia de <i>Libro</i> cuyos atributos <i>nombre</i> e <i>ISBN</i> tienen los valores <i>nl</i> e <i>isbn</i> respectivamente. - Por cada data <i>value</i> e en el conjunto <i>es</i>, existe en el sistema una nueva instancia de <i>ItemLibro</i> cuyo atributo <i>estado</i> tiene el valor <i>e</i>. El atributo <i>numEjemplar</i> tiene el valor del último número disponible para tal fin en el sistema (el cual se actualiza). - Se crea un link entre la nueva instancia de <i>Libro</i> y cada una de las nuevas instancias de <i>ItemLibro</i>.

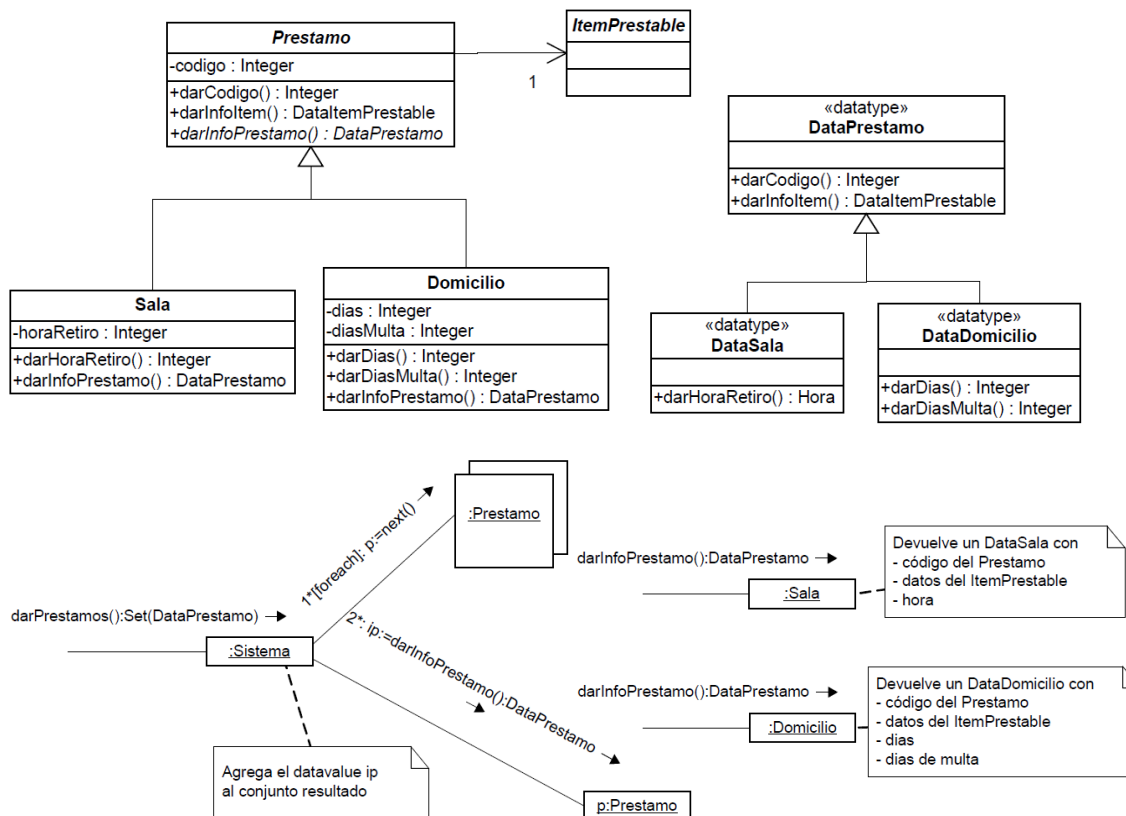
darPrestamosLibro(nl : String) : Set (DatosPrestamo)	
Descripción	Devuelve información de todos los préstamos de un libro.
Precondiciones	- Existe en el sistema una instancia de <code>Libro</code> identificada por <code>nl</code> .
Postcondiciones	Se retorna una colección de datavalues donde cada elemento: <ul style="list-style-type: none"> - Corresponde a una instancia de <code>Prestamo</code>, asociada a un <code>ItemLibro</code> correspondiente al <code>Libro</code> cuyo nombre es <code>nl</code>. - Tiene el nombre del <code>Usuario</code> asociado al <code>Prestamo</code>, la cantidad de días si es un préstamo a <code>Domicilio</code> y el número de ejemplar del <code>Libro</code>.

Se pide:

- Realizar los **Diagramas de Comunicación** de las dos operaciones descritas, indicando el tipo de visibilidad en todos los mensajes y definiendo el `DataType` `DatosPrestamo`.
- Realizar el **Diagrama de Clases de Diseño** resultante.

Problema 3 (25 puntos)

La biblioteca de su instituto ya realizó parte del diseño de un nuevo sistema de gestión de préstamos. La figura muestra un Diagrama de Clases de Diseño parcial, el cual contiene las clases relevantes para el subproblema aquí considerado. A su vez, el Diagrama de Comunicación presenta el diseño de la operación `darPrestamos`, que devuelve una lista de todos los préstamos existentes en el sistema.



Se pide:

- Implementar las declaraciones en C++ de las clases `Prestamo` y `Sala`.
- Implementar los constructores y destructores de las clases mencionadas en la parte i, e implementar la operación `darInfoPrestamo` donde corresponda (excepto en la clase `Domicilio`).
- Implementar en C++ la operación `darPrestamos`.