

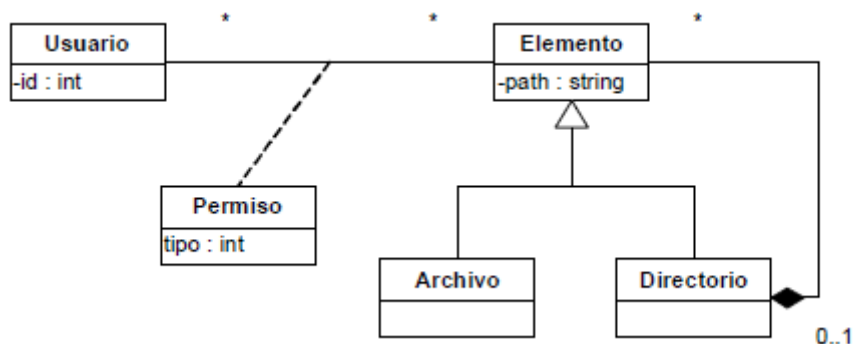
Segundo Parcial 2011

Presentar la resolución del parcial:

- Indique su nombre completo y número de cédula en cada hoja.
- Numere todas las hojas e indique la cantidad total de hojas que entrega en la primera.
- Escriba las hojas de un solo lado.
- **Comience cada ejercicio en una hoja nueva.**
- El parcial es individual y sin material. **APAGUE SU CELULAR.**
- **Escriba con lápiz y de forma prolija.**
- Duración: 2 horas.

Ejercicio 1 (30 puntos)

Se analizó un sistema de archivos que permite asignar permisos a usuarios del sistema operativo. El modelo de dominio realizado se puede ver en la figura siguiente. Los elementos del sistema de archivos pueden ser archivos o directorios, quien a su vez puede estar formado por otros elementos. Los elementos se identifican por su dirección absoluta (path). Sobre los elementos se definen permisos de acceso de cierto tipo que se asocian independientemente a cada usuario del sistema.



Además, se definieron dos operaciones que permiten asignar permisos y consultarlos posteriormente. Estas operaciones están definidas por los siguientes contratos reducidos.

Operación	altaDePermiso(idUsr:Integer, pathEl:String, tipoP:Integer)
Pre- y poscondiciones	
Pre: Existe una instancia de Usuario con id igual a idUsr.	
pre: Existe una instancia de Elemento con path igual a pathEl.	
pre: No existe una instancia del tipo asociativo Permiso con links a las instancias de Usuario y Elemento identificadas anteriormente. En caso de que la instancia de Elemento sea un Directorio, ninguno de sus Elementos están recursivamente linkeados con instancias de Permiso para ese mismo Usuario.	
post: Se creó una instancia de Permiso con tipo=tipoP y se asoció a las instancias de Usuario y Elemento identificadas en las precondiciones. Si el elemento era un Directorio entonces para todas las instancias de Elemento que lo componen se creó una instancia de Permiso como se definió antes, y así recursivamente.	

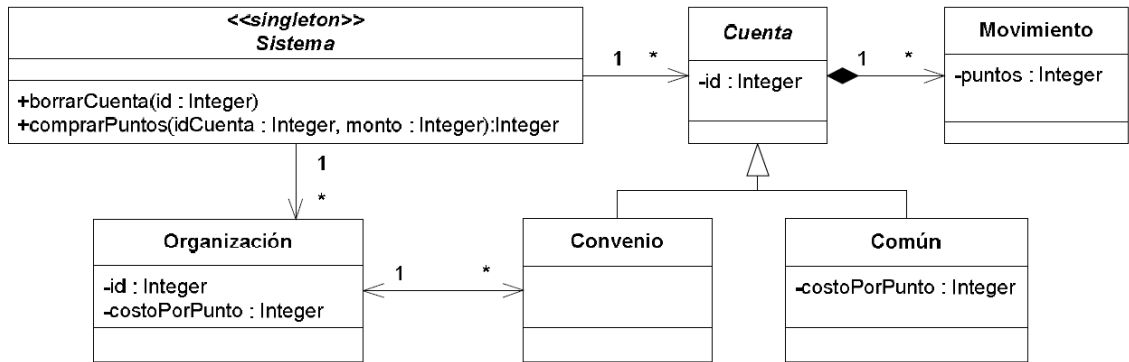
Operación	consultarPermisos(idUsr:Integer):Set(DataPermiso)
Pre- y poscondiciones	
<p>Pre: Existe una instancia de Usuario con id igual a idUsr.</p> <p>post: Para cada instancia de Elemento con el cual esté linkeado la instancia de Usuario, por medio del tipo asociativo Permiso, se define un datavalue de tipo DataPermiso que contiene el tipo de Permiso y path del Elemento relacionado con el Usuario. Se devuelve el conjunto de datavalues en cuestión.</p>	

Se pide:

- i) Realice los Diagramas de Comunicación para las operaciones especificadas en los contratos. No es necesario indicar las visibilidades.
- ii) Realice el Diagrama de Clases de Diseño completo de la solución.

Problema 2 (30 puntos)

Se le pidió que implemente un sistema para la gestión de las tarjetas de puntos de un supermercado. El sistema deberá registrar las cuentas de puntos de los clientes, los movimientos de puntos de cada cuenta (ganancia y uso de los puntos) y las organizaciones con la que el supermercado tiene convenios y promociones especiales. Se le ha entregado el siguiente diseño parcial que usted deberá respetar:



Se le ha entregado también las siguientes especificaciones para las operaciones pedidas:

Operación	Sistema::borrarCuenta(idCuenta : Integer)
Descripción	Elimina la instancia de Cuenta cuyo atributo id = idCuenta. Elimina también todas las instancias Movimiento asociadas. Si es una cuenta Convenio, desasocia a la cuenta de la Organización correspondiente.

Operación	Sistema::comprarPuntos(idCuenta : Integer, monto : Integer):Integer
Descripción	Crea una nueva instancia de Movimiento y la asocia a la cuenta cuyo atributo id = idCuenta. El valor del atributo puntos se calcula como el monto dividido el costo por punto de la cuenta. Si es una cuenta Común, el costo por punto de la cuenta es el valor del atributo costoPorPunto. Si es una cuenta Convenio, es el valor del atributo con el mismo nombre en la instancia de Organización asociada. El valor retornado es el valor del atributo puntos de la nueva instancia. Si el monto es menor que el costo por punto, no se crea el movimiento y se lanza una excepción.

Se pide:

- i) Implementar los .h de las clases Sistema, Cuenta, Común y Convenio.
- ii) Implementar en C++ las operaciones especificadas anteriormente y todas aquellas operaciones necesarias para su implementación que pertenezcan a las clases Sistema, Cuenta, Común y Convenio.