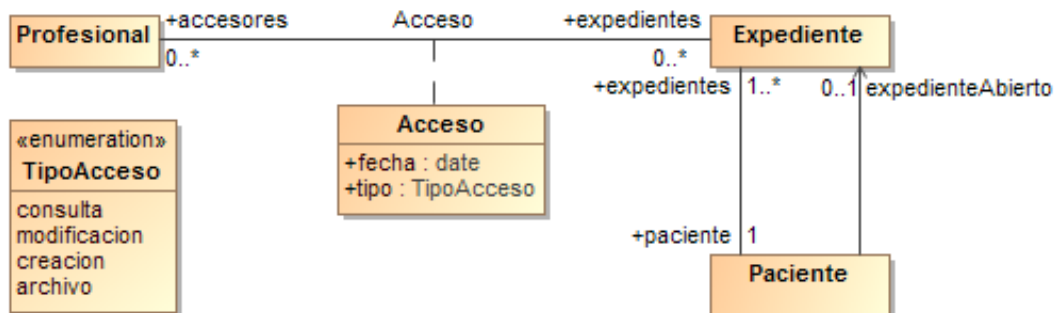


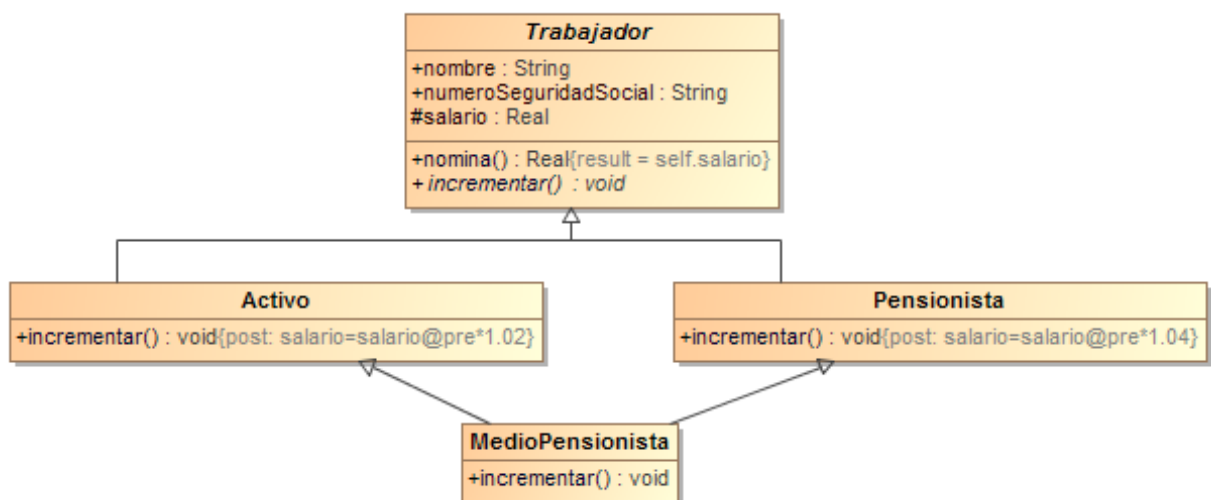
### Práctica 3.

**Ejercicio 1.** Supongamos un sistema para la gestión de expedientes médicos en un Hospital, descrito por el siguiente modelo que representa una relación (Acceso) entre los profesionales del hospital (médicos, enfermeras, personal administrativo, etc.) y los expedientes de los pacientes, de forma que se puedan conocer los diferentes accesos a cada expediente (quién lo hizo, cuándo, y para qué) y también conocer los expedientes a los que ha accedido un determinado trabajador del hospital.



Discutir las diferentes formas de diseñar un esquema del código de andamiaje Java necesario para implementar este modelo. Implementar en Java la decisión que se considere mejor en este caso, y justificar las decisiones que han llevado a elegir dicha estrategia de implementación.

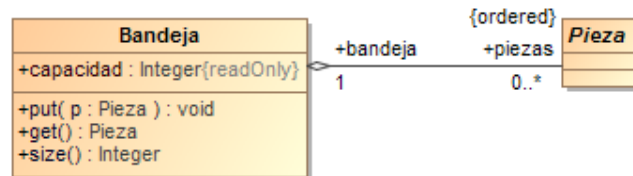
**Ejercicio 2.** El modelo estructural de la figura representa una jerarquía de clases para la elaboración de las nóminas de los trabajadores. La clase *Trabajador* contiene los atributos nombre, número de la seguridad social y salario, y los métodos nomina() e incrementar() (abstracto). Esta clase se especializa en *Activo* y *Pensionista*, puesto que en ambas categorías de trabajadores el cálculo de los incrementos anuales de salario se realiza de distinta forma, según se detalla en las correspondientes post-condiciones. El problema surge al aparecer la categoría de los MedioPensionista, cuya nómina se calcula a partir de la suma de su salario como activos y como pensionistas, precisando por tanto dos atributos "salario" que deben incrementarse cada uno como corresponde.



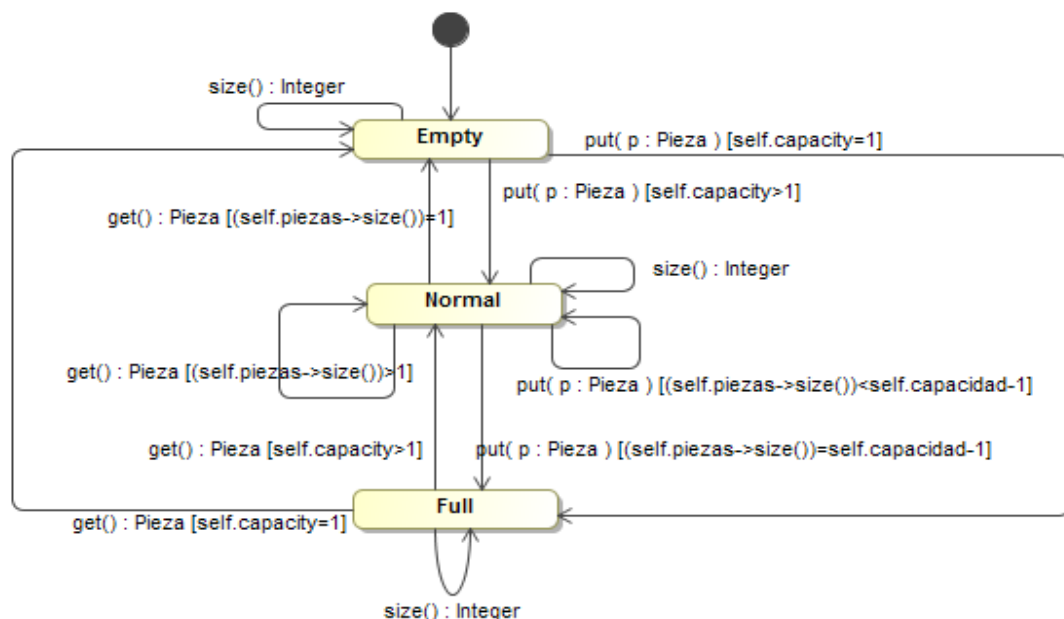
## Modelado y Diseño de Software

- Justificar por qué las clases descritas no pueden ser implementadas directamente en Java.
- Discutir y desarrollar una solución que permita resolver la situación descrita. Como es lógico, la solución propuesta debe mantener la funcionalidad actualmente existente en las tres subclases de Trabajadores y asegurar la consistencia de los atributos y la reutilización de los métodos de las clases Trabajador, Activo y Pensionista.
- Implementar en Java la solución propuesta.

**Ejercicio 3.** El ejemplo de la cadena de montaje de martillos planteaba el modelado de las bandejas que contenían las piezas de acuerdo al siguiente modelo.



Además, el diagrama de estados de una Bandeja distinguía los distintos estados por los que pasa una Bandeja y el efecto de las operaciones en cada uno de ellos.



Se pide implementar en Java dicho modelo de acuerdo al diagrama de estados, justificando la forma elegida para hacerlo de entre las tres posibles vistas en clase. Para ello, considerar la posibilidad de utilizar el **patrón de diseño Estado**, justificando las ventajas de esta solución frente a las otras posibles soluciones de diseño.