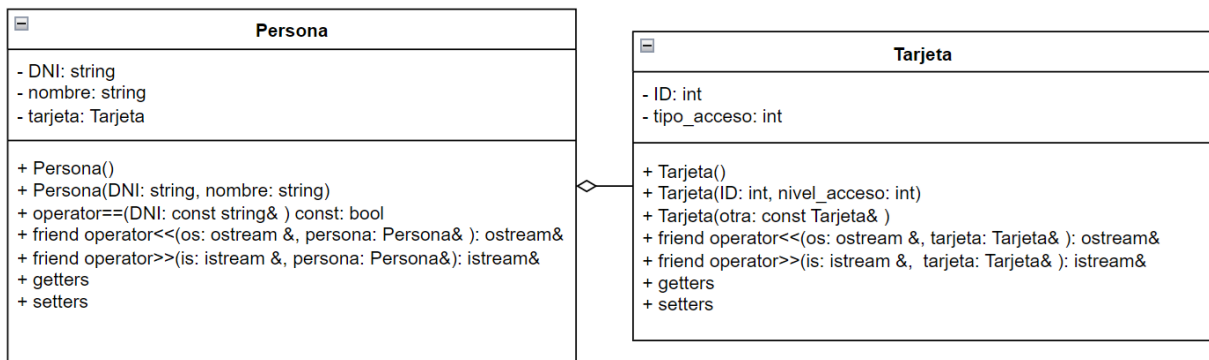


**Convocatoria Ordinaria.** Debe comprimir todos los ficheros que utilice y entregar solamente un único fichero llamado Apellido.Nombre.zip con su primer apellido y primer nombre. **Se corregirán solamente los archivos que se ejecuten sin errores.** Comente todo aquello que produzca errores. Documente los ficheros. Descargue los ficheros iniciales.

**Problema: (10.0 p)** El siguiente diagrama se ha utilizado para crear clases que permitan controlar el acceso a edificios.



Un edificio consta de 4 diferentes plantas y existen 4 niveles de acceso (0, 1, 2, 3). Cada nivel representa la máxima planta a la que puede ser accedida siempre que se posea una tarjeta con ese nivel:

1. Planta Baja: Recepción. Accesible a todas las personas que posean una tarjeta.
2. Primera planta: Recursos Humanos, TIC y Departamento de ventas. Accesible a todas las personas con nivel de acceso mayor o igual a 1.
3. Segunda planta: Gerenciales (CEO, CTO, etc.): Accesible a todas las personas con nivel de acceso mayor o igual a 2.
4. Tercera planta: Azotea (paneles solares, generadores, etc.). Accesible a todas las personas con nivel de acceso igual a 3.

Los datos de las personas se encuentran encolumnados dentro del fichero “personas.txt”, en donde cada fila almacena dos datos encolumnados separados por un espacio: DNI y nombre completo. Por lo tanto, un ejemplo puede ser el siguiente:

1533902M “Rafael Narvaez”

3506299L “Juan Maria Tena”

La base de datos de las tarjetas se encuentra dentro del fichero “tarjetas.pdf”, en donde cada fila almacena tres datos encolumnados separados por un espacio: ID, nivel\_acceso, y DNI. Los dos primeros corresponden a los atributos del objeto Tarjeta mientras que el tercero corresponde al

DNI de la persona asociada a la tarjeta. En este fichero tampoco existe cabecera. Por lo tanto, un ejemplo puede ser el siguiente:

16	2	9614567B
10	0	3567999L

Problema 1: Leer los datos de las personas y almacenar dentro de un contenedor de acceso aleatorio de memoria dinámica y almacenamiento eficiente llamado **personas**. Para ello, lea cada una de las líneas del fichero “personas.txt”, con excepción de la primera, y utilice estos datos para agregar objetos de la clase Persona al contenedor **personas**.

Problema 2: Leer los datos de las tarjetas y los DNIs asociados a cada tarjeta y almacenarlos dentro de dos contenedores del mismo tipo que **personas**: uno de objetos tipo Tarjeta llamado **tarjetas** y otro de strings llamados **DNI**s. Cada tarjeta y dni leído debe ser agregado de forma ordenada a los contenedores tarjetas y DNIs.

Problema 3: Asignar cada tarjeta a cada objeto Persona dentro de personas. Para ello, cree una iteración sobre los elementos de tarjetas y de los DNIs, simultáneamente ya que ambos tienen la misma cantidad de datos. A continuación, debe relacionar cada dni del contenedor DNIs con cada persona del contenedor **personas** utilizando la función `encontrarIndice(·)`, ésta recibe de entrada un contenedor de **personas** y un string `DNI_a_buscar` (que en cada iteración será el valor de cada elemento en DNIs). La función devuelve el índice del contenedor en el cuál se ha encontrado dicho DNI, si no se ha encontrado, se devuelve un valor de -1. Si el índice devuelto es diferente de -1, se debe modificar la tarjeta (`.setTarjeta(·)`) de la Persona en ese índice para que su atributo tarjeta sea igual a la tarjeta que se corresponde con dicho DNI (es decir, la tarjeta que se encuentra en la misma posición que el DNI buscado).

Problema 4. Cree un fichero de salida llamado “resumen.txt” el cuál almacene, sin ninguna cabecera, los atributos de todos los objetos Persona que se encuentran dentro del contenedor personas siguiendo el siguiente formato:

DNI	Nombre	ID	Nivel de Acceso
-----	--------	----	-----------------

Es decir, que un ejemplo de una línea podría ser:

7249169E	Enzo Olmo	11	1
----------	-----------	----	---

- Las cantidades mínimas de caracteres corresponden a: 10 para el DNI de la Persona, 25 para el nombre de la Persona, 3 para el ID de la Tarjeta y 3 para el nivel\_acceso de la Tarjeta.
- Las justificaciones del DNI de la persona y del nombre están justificadas a la izquierda, mientras que las justificaciones del ID de la tarjeta y del nivel\_acceso de la tarjeta son a la derecha.

(1.75 p) Llenar el primer contenedor con las personas que existen en el fichero “personas.txt”.

(2.25 p) Llenar los contenedores tarjetas y DNIs con los datos del fichero “tarjetas.pdf”.

(2.0 p) Asignar cada tarjeta a cada persona según los 3 contenedores obtenidos.

(1.50 p) Dar formato correcto de salida a los objetos Tarjeta y Persona.

(2.50 p) Generar el fichero de salida.