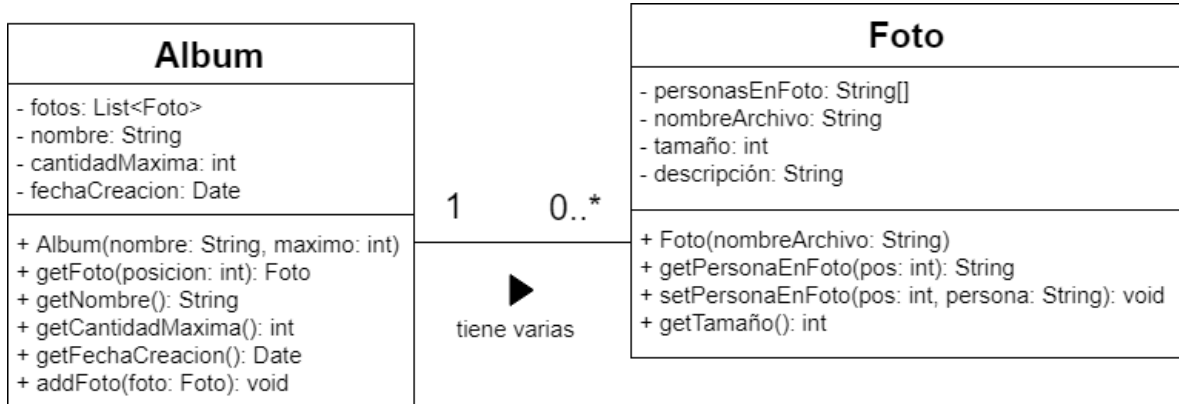


## Examen 111Mil – Administrador de Fotos

Durante el transcurso del dictado del programa 111Mil, los alumnos y profesores del programa tomaron fotografías de las actividades que realizaron durante el curso y las reuniones fuera del mismo. A uno de ellos se le ocurrió crear un sistema que permita administrar las fotos almacenadas, agrupandolas en álbumes con nombre: “Inicio de clases”, “Visita a una empresa”, “Juntada en Cervecería”, “Certificación”, etc.

El diseño del sistema es simple y se puede ver a continuación:



### Ejercicio 1. Implementar desde el diagrama de clases

La primera tarea en el desarrollo del administrador de fotos consiste en agregar funcionalidad básica a las clases. De acuerdo al siguiente código y al diagrama de clases anterior, complete:

- A. La declaración de las variables fotos, nombre y cantidadMaxima en la clase Album.
- B. El método addFoto de la clase Album, el cual agrega una foto nueva a la lista de fotos.
- C. El método getFoto de la clase Album, el cual obtiene una foto de la lista de fotos dada una posición en dicha lista.

```
public class Album {
    private Date fechaCreación = new Date();
    /*      * a) Completar declaración de variables (fotos, nombre
     * y cantidadMaxima).
     */
    private String nombre;
    private int cantidadMaxima;
    private List<Foto> fotos;

    public Album(String nombre, int maximo) {
        this.nombre = nombre;
    }
}
```

```

        this.cantidadMaxima = maximo;
        this.fotos = new ArrayList<Foto>();
    }

    /*
     * b) Agregar métodos getFoto y addFoto. Recordar que dichos
     * métodos utilizan la lista de fotos para obtener una Foto de una
     * posición y agregar una nueva foto a la lista.
     */

    public void addFoto(Foto foto) {
        this.fotos.add(foto);
    }

    public Foto getFoto(int posicion) {
        this.fotos.get(posicion);
    }

    public int getCantidadMaxima(){
        return this.cantidadMaxima;
    }

    public Date getFechaCreacion(){
        return this.fechaCreación;
    }

    public String getNombre(){
        return this.nombre;
    }
}

```

## Ejercicio 2. Implementar un método a partir de un enunciado

Luego de unos meses, el sistema de administración de fotos almacena un número considerable de álbumes y fotos. Muchas de esas fotos fueron almacenadas de forma errónea, por ejemplo, en mala resolución o con errores de guardado. A una de la alumnas del programa se le ocurre crear un método en Album que permita saber si en el mismo hay fotos con tamaño menor a cierto valor. Por ejemplo, si una foto está mal guardada, se puede identificar porque tiene tamaño 0. El método debe tener la siguiente signatura:

```

public boolean tieneFotoConTamañoMenor(int máximo){
    // Completar
}

```

```

    for (Foto fotoActual : fotos)
    if (fotoActual.getTamaño() <= max)
        return true;
    return false;
}

```

### Ejercicio 3. Determine la opción correcta.

El profesor del curso también realizó su aporte al administrador de fotos, agregando un método nuevo al Album. Sin embargo, no documentó el método agregado intencionalmente, para que sus alumnos intenten descubrir qué funcionalidad realiza. Hasta este momento, los alumnos descubrieron 4 posibilidades, indique cuál de ellas es la correcta:

```

public List<Foto> nuevaFuncionalidad(int maximo) {
    List<Foto> r = new ArrayList<Foto>();
    for (Foto fotoActual : this.fotos)
    if (fotoActual.getTamaño() <= max) {
        int posInsert = 0;
        boolean encontrado = false;
        while (posInsert < r.size() && !encontrado) {
            Foto fotoListaRetorno = r.get(posInsert);
            if (fotoActual.getTamaño() > fotoListaRetorno.getTamaño())
                encontrado = true;
            else
                posInsert = posInsert + 1;
        }
        r.add(posInsert, fotoActual);
    }
    return r;
}

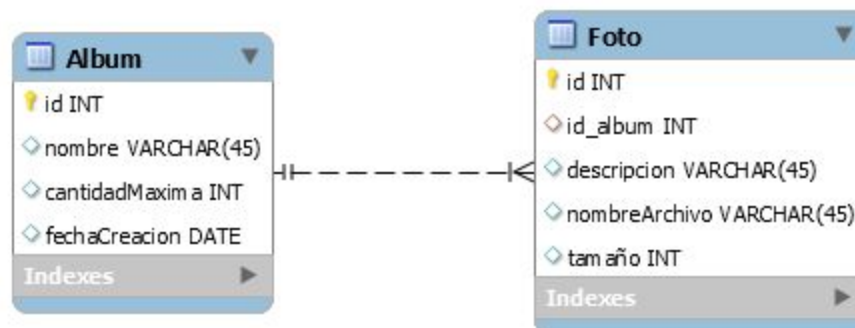
```

A	Filtra las fotos con tamaño mayor a un valor dado y luego busca la foto con tamaño más pequeño.
B	Filtra las fotos con tamaño menor a un valor dado y las retorna ordenadas de mayor tamaño a menor tamaño.
C	Retorna la/las fotos con mayor tamaño en el álbum.
D	Ninguna de las anteriores es correcta.

Respuesta: **B**

#### Ejercicio 4. Interpretación de Esquema de Tablas.

Dado el diagrama de entidades y relaciones del administrador de fotos:



Indique si las siguientes afirmaciones son verdaderas o falsas.

La relación entre Álbum y Foto es N a N	F
Cada Foto puede pertenecer, como máximo, sólo a un álbum.	V
La clave primaria de Álbum es id.	V
El valor "20 MB" es un valor válido para el tamaño de la foto.	F
100 es un valor posible para cantidadMaxima en la tabla Álbum.	V
El álbum puede, como máximo, tener asociada solo una foto.	F
La clave principal de Foto es id_album.	F
La tabla Foto tiene una clave compuesta	F

#### Ejercicio 5. Consultas SQL.

El disco donde se almacenan las fotografías del sistema está a punto de alcanzar su capacidad máxima. Por este motivo, uno de los alumnos quiere consultar en la base de datos aquellas fotografías con tamaño mayor a 1024, ordenadas de mayor tamaño a menor tamaño. Escriba la consulta SQL que permita obtener el **nombre de archivo, el tamaño y el nombre del álbum de aquellas fotos que cumplen con tamaño mayor o igual a 1024, ordenadas por tamaño de forma descendente.**

```
SELECT f.nombreArchivo, f.tamaño, a.nombre FROM Foto f
JOIN Album a ON a.id = f.id_album
WHERE f.tamaño >= 1024
ORDER BY f.tamaño DESC
```

